Visual Analytics Approaches for Descriptor Space Comparison and the Exploration of Time Dependent Data

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

DISSERTATION

zur Erlangung des akademischen Grades eines
Doktor-Ingenieurs (Dr.-Ing.)
von

Dipl. Bioinf. Sebastian Bremm

geboren in Frankfurt, Deutschland

Referenten der Arbeit: Prof. Dr. techn. Dieter W. Fellner
Technische Universität Darmstadt
Prof. Dr. Tobias Schreck
Universität Konstanz

Tag der Einreichung: 19.09.2013
Tag der mündlichen Prüfung: 02.12.2013

Darmstadt 2013
D 17

Erklärung zur Dissertation

Hiermit versichere ich die vorliegende Dissertation selbständig nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 21.10.2013 Sebastian Bremm

1. Zusammenfassung

Sensorik, Robotik und verbesserte Messverfahren führen zu einer Schwemme an Beobachtungsdaten, deren manuelle Auswertung den Benutzer überfordert. Die computergestützte Auswertung ermöglicht einen effektiveren Umgang mit diesen Datenmengen. Dabei können zwei Analyseziele unterschieden werden. In einigen Szenarien existieren klar definierte Muster die gefunden, oder Hypothesen, die überprüft werden sollen. Hier erzielen komplett automatisierte Ansätze sehr gute Resultate, da sie wesentlich effizienter als der Mensch eindeutig beschriebene Muster in großen Datenmengen finden können. Im Gegensatz dazu steht die explorative Datananalyse [Tuk77] mit der ich mich in dieser Arbeit beschäftige. Dabei hat der Analyst kein oder nur wenig Wissen über etwaig vorhandene Muster oder Zusammenhänge. Üblicherweise existieren zwar Annahmen über die Daten, deren Überprüfung ist aber nicht alleiniges Ziel der Analyse. Thomas und Cook prägten den Leitsatz: „Detect the expected and discover the unexpected“ - „Finde das Erwartete und entdecke das Unerwartete“ [TC05]. Um neue, unerwartete Muster zu finden und deren Relevanz zu beurteilen, muss der Benutzer in den Analyseprozess integriert werden.

Visual Analytics ermöglicht dem Nutzer eine explorative Analyse großer Datenmengen durch die Kombination von Visualisierung und Data Mining. Durch Informationsvisualisierung können komplexe Datenmengen anschaulich dargestellt werden. Dadurch können Zusammenhänge und Muster vom Analysten erkannt werden. Sehr große Datensätze überfordern aber schnell die kognitiven Fähigkeiten eines Menschen. Daher werden bei den Visualisierungen oft Ansätze verfolgt, die die Daten in verschiedene semantische Gruppen unterteilen und je nach Ansatz immer nur einen ausgewählten Teil darstellen. Beispiele für solch grundlegenden Ansätze der Informationsvisualisierung sind unter anderem „overview first, zoom and filter, then details-on-demand“ - „Biete zunächst eine Übersicht, Zooming und Filterung, danach Details auf Anfrage“ [Shn96] oder „Focus and Context“ [CMS99]. Eine automatische Datenanalyse kann helfen, die Daten besser aufzubereiten oder potentiell interessante Strukturen in den Daten zu identifizieren. Dadurch wird dem Benutzer ein Teil der Aufgaben abgenommen und er kann seine menschlichen Qualitäten, das Erkennen von unbekannten Mustern und die Bewertung gefundener Strukturen, besser in den Analyseprozess einbringen.

Dieser Analyseprozess ist keine Einbahnstraße, sondern eher ein mit Schleifen kombiniertes Vor und Zurück. Beispielsweise könnte ein Analyst, nach initialer Aufbereitung und Visualisierung der Daten, ein ungewöhnliches Muster finden. Daraufhin können die Parameter der automatischen Analysealgorithmen angepasst und optimiert werden um das Muster besser erkennen und einordnen zu können. Das daraus gewonnene Wissen kann dann zur Formulierung neuer Hypothesen im nächsten Analysedurchgang dienen.

Die Idee dieser engen Kombination von automatischen Analysealgorithmen, interaktiver Visualisierung und Integration von Expertenwissen dient als Basis für die beiden folgenden Teile meiner Arbeit, den visuell-interaktiven Vergleich von Deskriptoren (siehe Kapitel I) und die visuelle Analyse von zeitabhängigen Daten (siehe Kapitel II). Obwohl sich beide Bereiche diese Kombination zu eigen machen, unterscheiden sich die jeweils verwendeten Methoden, da sie stark von den jeweiligen Anwendungsfällen und Charakteristka der Daten abhängen. So kann es besispielsweise in beiden Bereichen hifreich sein, Gruppen von Objekten zu identifiezieren. Die Ansätze zum Clustering und der Visualisierung der Ergebnisse ähneln sich, weisen aber auch signifikante Unterschiede auf da die Integration des Zeitbezuges häufig eine Anpassung der Vorgehensweise erfordert.

Visueller Vergleich von Deskriptoren

Um große Datenmengen aus komplexen Objekten wie Netzwerken oder Molekülen computergestützt verarbeiten zu können, wird für diese eine kompakte Repräsentationsform benötigt. Diese sollte für die Analyse relevante Information enthalten. Solche Beschreibungen können sowohl einfache numerische Werte sein, wie etwa die Ladung eines Moleküls, aber auch lange Ketten von Zeichen, die zum Beispiel die DNA Struktur eines Bakteriums repräsentieren. Jede dieser Beschreibungen stellt einen Deskriptor dar, die ihrerseits beliebig zu sogenannten Deskriptorräumen kombiniert werden können. Durch die explorative Analyse und den Vergleich verschiedener Repräsentationen der Daten können Muster und Zusammenhänge sichtbar werden bzw. deren Bewertung erleichtert werden. Ziel der Analyse ist es Ähnlichkeiten und Unterschiede zwischen verschiedenen Deskriptoren sowohl auf globaler als auch lokaler Ebene zu finden. Das gilt sowohl im paarweisen Vergleich als auch im Vergleich aller verfügbarer Deskriptoren. Tritt ein Muster etwa in verschiedenen Deskriptoren der gleichen Daten auf, ist es potentiell interessanter und verlässlicher als bei nur seltenem Auftreten. Ich stelle in dieser Arbeit Ansätze für den Vergleich von Daten vor, deren Struktur hierarchisch oder nicht bekannt ist. In beiden Fällen folge ich dem in der Visual Analytics typischen Ansatz und verbinde eine automatische Analyse, zum Beispiel die algorithmische Bewertung von Ähnlichkeit, mit einer interaktiven Visualisierung um das Wissen des Benutzers in die explorative Anaylse zu integrieren.

Visueller Vergleich von hierarchisch strukturierten Daten

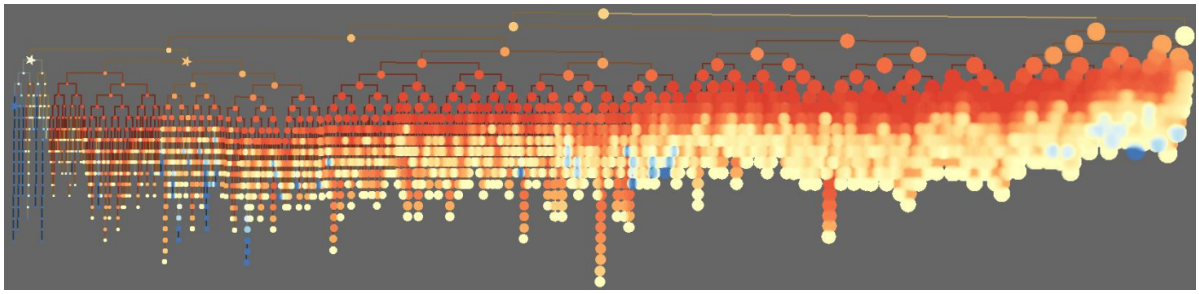


Figure 1.1.: CloudTree Visualisierung eines Vergleichbaums mit über 7000 Elementen (siehe Kapitel 5).

Hierarchisch strukturierte Daten treten in vielen Analyseszenarien auf. Entweder sind die Daten an sich (implizit) hierarchisch strukturiert, wie etwa phylogenetische Bäume, in denen der evolutionäre Prozess abgebildet wird oder die Daten werden in einem Datenvorverarbeitungsschritt in diese Form gebracht. Dies geschieht etwa durch hierarchisches Clustering, das im Gegensatz zu anderen Gruppierungsverfahren den Vorteil hat, a priori keine Annahmen über die Anzahl der Cluster oder die Struktur der Daten zu treffen. Verfahren, die auf den Vergleich solcher Hierarchien (im Folgenden auch Bäume genannt) spezialisiert sind, können spezifische Vorteile dieser Strukturen sowohl für eine Verbesserung der Laufzeit, etwa bei Suchen, oder die Visualisierung etwa durch Node-Link Diagramme nutzen.

Im Rahmen dieser Arbeit habe ich eng mit Biologen zusammengearbeitet, um die spezifischen Fragestellungen zu analysieren und eine bedarfsgerechte Lösung zu erarbeiten. Phylogenetische Bäume beschreiben die evolutionäre Verwandschaft der enthaltenen Spezies. Jedes Blatt repräsentiert einen Organismus, die inneren Knoten stellen die berechneten hypothetischen Vorfahren dar. Die formale Repräsentation der einzelnen Organismen spielt bei der Analyse eine zentrale Rolle. Üblicherweise werden dafür DNA-, RNA- oder Aminosäuresequen-

zen von Proteinen herangezogen. Dabei bietet sich zum Beispiel RNA- und Proteinuntereinheiten des Ribosoms an, da dieses in allen Organismen vorhanden ist. Die Untereineheit stellen Deskriptoren dar, deren jeweiligen Auswirkung auf den resultierenden phylogenetischen Baum unklar sind.

Um die Bäume vergleichen zu können, habe ich eine **neues Bewertungsverfahren der Ähnlichkeit** von (Sub-)Bäumen vorgeschlagen. Im Gegensatz zu existierenden Ansätzen werden nicht nur die Blätter sondern alle Knoten bei der Berechnung berücksichtigt. Dadurch gelingt eine Integration der Baumstruktur in das Ähnlichkeitsmaß. Sowohl der blattbasierte als auch mein strukturbasierter Vergleichsscore ergeben aufgrund gleicher Blätter in allen Bäumen trotz maximaler Strukturunterschiede einen Wert >0. Um an dieser Stelle das intuitive Verständnis der Ähnlichkeit zu verbessern, stelle ich ein **neues Normalisierungsschema** vor.

Für die Darstellung und Analyse der Bäume und ihrer Ähnlichkeit habe ich das **interaktives Visualisierungstool „ViPhy“** entworfen. Im Vorfeld wurde in Zusammenarbeit mit Biologen eine Liste potentieller Analyseziele erarbeitet, die das Programm unterstützt. Es kombiniert unterschiedliche Ansichten, um die verschiedenen Aspekte der explorativen Analyse abzudecken. Unterschiedliche Detailgrade erlauben sowohl eine Übersicht über die globale Ähnlichkeit aller Bäume sowie genaue 1:n oder detaillierte paarweise Vergleiche. Dadurch können zum Beispiel auch interessante lokale Muster oder Subbäume, die in allen Bäumen identisch vorkommen schnell identifiziert werden.

Die Visualisierungslösung wurde anschließend anhand eines echten Datensatzes mit 35 Bäumen aus jeweils 32 Bakterienarten getestet. Auf globalem Level konnte schnell eine Unähnlichkeit der Bäume festgestellt werden, obwohl alle berechneten Phylogenien auf verschiedenen Untereinheiten des gleichen Proteins beruhen. In der weiteren Analyse konnten einige sehr stabile Subbäume identifiziert werden, die in fast allen Bäumen existieren. Dieses ungewöhliche Muster konnte nicht intuitiv erklärt werden und dient als Startpunkt für weitere biologische Forschung.

Um die Skalierbarkeit des Systems zu erhöhen, habe ich eine **neue Art der Baumvisualisierung** entworfen. Bei der klassischen Knoten/Kanten Darstellung von Bäumen kommt es zu Überlappungen innerhalb der Visualisierung. Für die Darstellung tausender Blätter bräuchte man mindestens eben soviele horizontal nebeneinander liegende Pixel, um diesesn Effekt zu vermeiden. Da die Überlappung bei der parallelen Darstellung mehrerer großer Bäume nicht zu vermeiden ist, habe ich eine Visualisierungstechnik vorgeschlagen, die trotz großer Überlappungen auch kleine Details innerhalb der Bäume darstellen kann.

In den sogenannten CloudTrees werden bei Darstellung eines Pixels alle damit abgebildeten Knoten miteinander verrechnet. Dabei kann die Gewichtung der einzelnen Elemente dem Analyseszenario dynamisch angepasst werden. So können etwa Knoten mit einem hohen Ähnlichkeitsscore stärker bei der Darstellung berücksichtigt werden. Dadurch können auch potentiell interessante, sehr kleine Subbäume sichtbar gemacht werden. Des Weiteren wurden die Interaktionsformen angepasst, um deren Skalierbarkeit gleichermaßen zu gewährleisten. Wenn ein einzlener Pixel mehr als ein Element des Baumes repäsentiert, ist eine eindeutige Auswahl eines spezifischen Elementes durch Selektion des Pixels nicht mehr ohne Weiteres möglich. Dem Analysten werden daher verschiedene Selektionsschemata zur Verfügung gestellt, etwa um den besten, schlechtesten oder größten Subbaum des selektierten Pixels zu wählen. Durch die Kombination der von mir vorgestellten Ansätze können mehrere große hierarchische Datensätze effizient sowohl auf globaler als auch auf lokaler Ebene miteinander verglichen werden.

Visueller Vergleich von Daten mit unbekannter Struktur

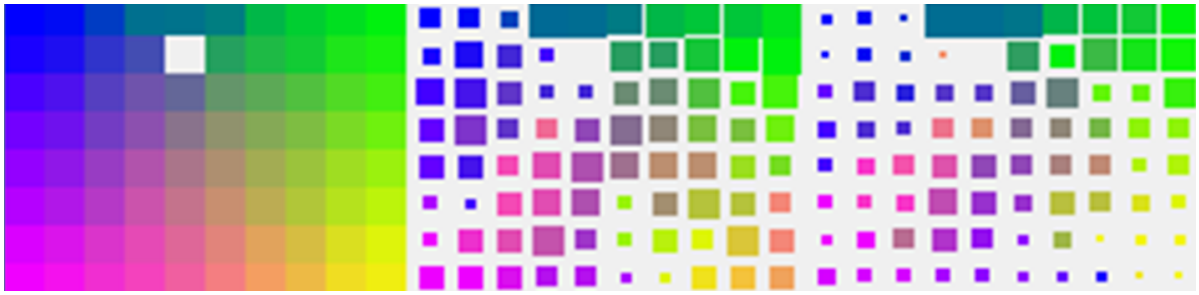


Figure 1.2.: Vergleich einer Referenz Self Organizing Map (links) mit zwei anderen SOMs unterschiedlicher Ähnlichkeit (mitte & rechts) (siehe Kapitel 6).

Der Vergleich von Daten mit unbekannter Struktur verfolgt im Allgemeinen die gleichen Ziele wie der Vergleich hierarchischer Daten. Es sollen globale und lokale (Un-)Ähnlichkeiten identifiziert und mit den restlichen Daten in Zusammenhang gebracht werden können. Häufig liegen nicht hierarchisch organisierte Daten vor und teilweise ist die künstliche Herbeiführung einer Hierarchisierung, etwa durch hierarchisches Clustering nicht erwünscht, da dadurch immer ein durch den Algorithmus verursachter Bias mit in die Analyse einfließen kann. Da also die Hierarchie nicht als Basisstruktur für die Ähnlichkeitsberechnung der Deskriptoren dienen kann, müssen andere Metriken herangezogen werden. Diese sollten unabhängig von der Dimensionalität und dem Datentyp des jeweiligen Deskriptors berechenbar sein.

Ich schlage vor, die **Nachbarschaftsbeziehungen der einzelnen Objekte als Grundlage zur Bewertung der Ähnlichkeit von Deskriptoren** zu verwenden. Darauf basierdende Maße werden häufig zur Evaluierung der Projektionsgüte von Dimensionsreduktions-Algorithmen verwendet. Ich habe verschiedene topologie- und distanz basierte eigene und aus der aktuellen Literatur entnommenen Maße implementiert und miteinander verglichen. Um diese zu testen, habe ich synthetische Daten mit dem von mir dafür entworfenen interaktiven Visualisierungswerkzeug PCDC (Parallel Coordinates Data Creation - Datengenerierung basierend auf Parallelen Koordinaten) erzeugt. PCDC erlaubt eine schnelle Definition von hochdimensionalen Verteilungen verschiedener Klassen. Dadurch konnten gezielt unterschiedliche Szenarien, etwa mit varierendem Rauschen oder Gruppenanzahl, simuliert werden.

Für den visuellen Vergleich von Daten mit unbekannter Struktur stelle ich ein auf zweidimensionalen Farbkarten beruhendes Verfahren vor. Es erlaubt die schnelle Erkennen von globaler und lokaler Ähnlichkeit. Für die Darstellung müssen die Daten zunächst, wie bei vielen Visualisierungen üblich, in den zweidimensionalen Raum projetziert werden. Dies kann mittels geeigneter Projektionsverfahren wie zum Beispiel der Hauptkomponentenanalyse (Principal Component Analysis, PCA), der Multidimensionalen Skalierung (Multidimensional Scaling, MDS) oder durch Selbsorganisierende Karten (Self-Organizing Maps, SOMs) geschehen. Das Resultat sind zweidimensionale Punktwolken (PCA, MDS) oder ein zweidimensionales Raster (SOM). Werden zwei Deskriptoren miteinander verglichen, so wird die Kodierung der Position einer der beiden Projektionen durch eine zweidimensionale Farbkarte übernommen. Diese Farbcodierung kann nun auf die jeweiligen Objekte in der 2D Darstellung der verglichenen Projektion übertragen werden.

Wenn die beiden verglichenen Projektionen ähnlich sind, ergibt sich ein homoger Farbverlauf. Sind sie unähnlich, ist kein Farbverlauf zu erkennen. Lokale Ähnlichkeit wird durch lokale Farbhomogenität sichtbar. Das Verfahren wird auf die Rasterdarstellung der SOM erweitert, bei der die Daten gleichzeitig durch den Algo-

rithmus gruppiert wurden. Jedes Feld kann mehrere Eingabeobjekte enthalten, wodurch eine Skalierbarkeit des Verfahrens gewährleistet wird. Darauf aufbauend wird die Visualisierung um weitere Aspekte wie zum Beispiel eine Darstellung des Visualisierungsfehlers erweitert.

Pipeline für den visuellen Vergleich von Deskriptoren

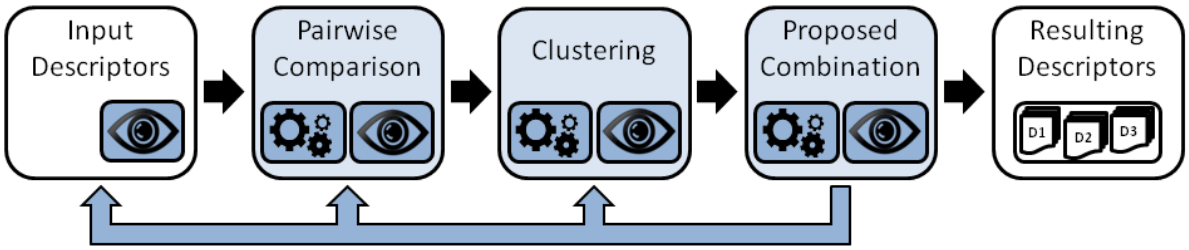


Figure 1.3.: Pipeline für den visuellen Vergleich von Deskriptoren (siehe Kapitel 7)

Obwohl sich die konkrete Umsetzung beim Vergleich von Daten mit hierarchischer oder unbekannter Struktur zum Beispiel in den Verwendeten Distanzmaßen unterschiedet, gibt es signifikante Ähnlichkeiten im generellen Vorgehen. Ich schlage daher eine generische Pipeline für die vergleichende Analyse von Deskriptoren vor.

Um die Einstiegshürde für den Benutzer so gering wie möglich zu halten, wird zunächst in einem automatischen Durchlauf ein initialer Ergebnisvorschlag berechnet. Ähnliche Deskriptoren werden zu Gruppen zusammengefasst und jeweils durch einen Repräsentanten dargestellt. Dies dient dem Nutzer als Einstieg in die Analyse. Ihm steht es nun frei, nur mit den wenigen Repräsentanten weiter zu arbeiten oder tiefer in den Vergleichsprozess einzusteigen. Dabei werden die einzelnen Schritte, paarweise Ähnlichkeitsberechnung, Gruppierung und Repräsentantenbestimmung durch interaktive Visualisierungen unterstützt. Der Benutzer kann sowohl die Daten explorativ analysieren als auch neue Parameter für die einzelnen Berechnungen setzen und so sein Wissen in den Prozess einbringen.

Analyse zeitabhängiger Daten

Zeitabhängige Daten treten in vielen Anwendungsgebieten wie der Finanazanalyse, der Klimabeobachtung oder der Medizin auf. Die zeitliche Komponente spielt dabei eine entscheidende Rolle, die sowohl bei der automatischen Analyse als auch bei der Visualisierung berücksichtigt werden muss. Viele Methoden ähneln stark denen aus anderen Spezialisierungsfeldern innerhalb der Visual Analytics, sind aber auf die spezifischen Anforderungen zeitabhängiger Daten angepasst. Diese Arbeit enthält Beiträge zu zwei Teilbereichen, der Analyse gruppierter zeitabhängiger Daten und der Analyse sequenzieller zeitabhängiger Daten.

Explorative Analyse gruppierter Daten mit Zeitbezug

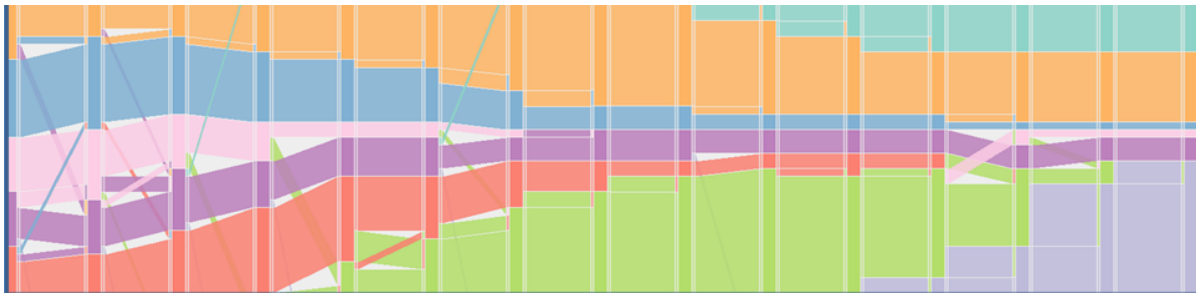


Figure 1.4.: Visualisierung der Gruppenstruktur und deren Änderung Ãijber die Zeit (siehe Kapitel 9).

Bei gruppierten Daten ist jedem beobachteten Datenpunkt zu jedem Zeitpunkt eine eindeutiges Gruppenlabel zugeordnet. Diese Gruppenzugehörigkeit kann sich über die Zeit ändern. Ziel der Analyse ist es, solche Änderungen zu identifizieren und verfolgen zu können. Dabei ist sowohl die Beobachtung einzelner Objekte als auch die Betrachtung globaler Dynamiken von Bedeutung. Mein Lösungsansatz kombiniert eine automatische Analyse der Daten mit einer interaktiven Visualisierung.

Die Visualisierung lehnt sich an der Darstellung paralleler Sets [KBH06] an. Jeder Zeitpunkt wird durch einen Balken repräsentiert, der farblich in die vorhandenen Gruppen unterteilt ist. Die größe einer Gruppe, definiert durch die Anzahl an enthaltenen Objekten, wird durch die größe der jeweiligen Farbpartition repräsentiert. Kanten zwischen den Balken zweier Zeitpunkte veranschaulichen die Gruppenwechsel oder Stabilität der Objekte. Auch hier spiegelt die Größe der jeweiligen Kante die Menge der repräsentierten Objekte wieder. Der Benutzer kann einzelne Kanten oder Gruppen zu einem bestimmten Zeitpunkt auswählen. Dadurch werden die enthaltenen Objekte über alle Zeitpunkte hinweg hervorgehoben und können so im globalen zeitlichen Kontext analysiert werden.

Wenn lange Zeitspannen analysiert werden sollen, muss eine Auswahl der darzustellenden Zeitpunkte getroffen werden. **Ich habe verschiedene neue Verfahren vorgestellt, um potentiell interessante Zeitpunkte zu identifizieren**. Diese lassen sich in zwei Kategorien unterteilen. Zum einen werden repräsentative Zeitpunkte so ausgewählt, dass die Anzahl der dazwischen liegenden Gruppenwechsel konstant ist. Dadurch erhält der Nutzer einen Überblick über die globalen Charaktersitika der Daten. Zum anderen können automatisch Zeitpunkte selektiert werden, die Besonderheiten wie seltene Gruppenwechsel beinhalten.

Das System wurde auf zwei Szenarien angewand. Im ersten Fall wurden Bewegungsmuster bei einer Evakuierung nach einem fiktiven Bombenanschlag analysiert. Die Daten hierzu stammen aus der IEEE VAST Challenge 2008 und enthalten die Bewegungsprofile von 82 Personen zu 837 Zeitpunkten [AA10]. Es wur-

den 20 automatisch selektierte, repräsentative Zeipunkte visualisiert. Ab dem gut zu identifizierenden Zeitpunkt der Bombenexplosion sind klare Bewegeunsmuster zu den Notausgängen zu erkennen. Es konnten aber auch deutlich einige Personen identifiziert werden, die nicht den vorgesehen kürzesten Wegen folgen und nicht rechtzeitig das Gebäude verlassen konnten. Im zweiten Anwendunsfall wurden die Telefondaten von 5.108.298 Mobilfunkverbindungen von 367.730 Personen über neun Monate rund um Mailand analysiert. Dies geschah in Zusammenarbeit mit Kollegen aus dem Fraunhofer ISAS Bonn, deren Geoanalyse Software zu diesem Zwecke mit meinem Ansatz kombiniert wurde. Dadurch konnten verschiedene Verhaltensmuster der beobachteten Personen, wie etwa der Besuch des Flohmarktes an Sonntagen, identifiziert werden.

Explorative Analyse zeitlicher Sequenzreihen

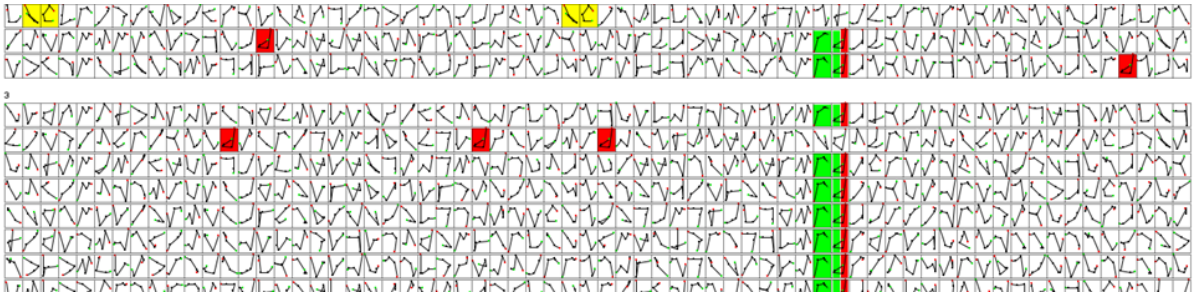


Figure 1.5.: Vergleichsvisualisierung verschiedener Finanzreihen. Während der Exploration ausgewählte Muster sind hervorgehoben (siehe Kapitel 10).

Um hochdimensionale oder sehr lange Zeitreihen zu analysieren, können diese durch Sequenzen von abstrakten Symbolen kodiert werden. Jedes Symbol beschreibt dabei ein oder mehrere zeitabhängige Muster. Dadurch kann die Komplexität und Länge des ursprünglichen Datensatzes reduziert werden. Ziel der Analyse ist es üblicherweise signifikante Muster, etwa ähnliches Verhalten mehrer Objekte zum gleichen Zeitpunkt zu identifizieren. Hierfür muss allerdings sowohl Signifikanz als auch Ähnlichkeit definiert oder bewertet werden. Der vorgestellte Lösungsansatz bezieht dazu den Benutzer in die Analyse mit ein. So kann die Quantifizierung der Ähnlichkeit zweier Symbole durch den Benutzer interaktiv paarweise oder durch Definition exemplarischer Muster definiert werden.

Die darauf aufbauende Analyselösung basiert auf einer **Kombination unterschiedlicher Visualisierungen, die verschiedene Aspekte der Daten betrachten**. Die Sequenzansicht bietet einen Überblick über alle Serien. Diese können nach verschiedenen Kriterien wie etwa selektierten Subsequenzen oder zusätzlichen Metadaten sortiert werden. Verschiedene Heatmaps zeigen die Häufigkeit von Subsequenzen über alle Zeitpunkte bzw. über alle Sequenzen und mit Hilfe interaktiver Streudiagramme können Zusammenhänge ausgesuchter Variablen untersucht werden. Dies können sowohl Metadaten als auch speziell für die Sequenzanalyse berechnete Deskriptoren wie zum Beispiel die mittlere Häufigkeit des Musters pro Sequenz sein. Ein prototypischer Analysevorgang wird anhand eines finanzwirtschaftlichen Datensatzes dargelegt. Durch deren Verknüpfung, etwa der Hervorhebung selektierter Subsequenzen in allen Ansichten, können die einzelnen Stärken kombiniert werden und so auch komplexe Datensätze und Fragestellungen gelöst werden. Exemplarisch wurde ein Datensatz aus der Finanzwirtschaft mit den Risiko und Ertragseinschätzungen von 60 Firmen über 500 Tage analysiert. Dabei konnten interessante Muster wie etwa ein ähnliches Verhalten mehrer Firmen innerhalb von zwei Wochen identifiziert werden.

Zusammenfassend kann gesagt werden, dass durch die Integration des Benutzers in die Analyse wertvolles Wissen gewonnen wird. Durch die aufeinander abgestimmte Kombination von automatischer Analyse und interaktiver Visualisierung können auch große und komplexe Datensätze verarbeitet werden. Ich konnte dies an Hand verschieder Ansätze in unterschiedlichen Einsatzbereichen zeigen.

2. Abstract

Modern technologies allow us to collect and store increasing amounts of data. However, their analysis is often difficult. For that reason, Visual Analytics combines data mining and visualization techniques to explore and analyze large amounts of complex data. Visual Analytics approaches exist for various problems and applications, but all share the idea of a tight combination of visualization and automatic analysis. Their respective implementations are highly specialized on the given data and the analytical task. In this thesis I present new approaches for two specific topics, visual descriptor space comparison and the analysis of time series.

Visual descriptor space comparison enables the user to analyze different representations of complex datasets e.g., phylogenetic trees or chemical compounds. I propose approaches for data sets with hierarchic or unknown structure, each combining an automatic analysis with interactive visualization. For hierarchically organized data, I suggest a novel similarity score embedded in an interactive analysis framework linking different views, each specialized on a particular analytical tasks. This analysis framework is evaluated in cooperation with biologists in the area of phylogenetic research. To extend the scalability of my approach, I introduce CloudTrees, a new visualization technique for the comparison of large trees with thousands of leaves. It reduces overplotting problems by ensuring the visibility of small but important details like high scoring subtrees.

For the comparison of data with unknown structure, I assess several state of the art projection quality measures to analyze their capability for descriptor comparison. For the creation of appropriate ground truth test data. I suggest an interactive tool called PCDC for the controlled creation of high dimensional data with different properties like data distribution or number and size of contained clusters. For the visual comparison of unknown structured data, I introduce a technique which bases on the comparison of two dimensional projections of the descriptors using a two dimensional colormap. I present the approach for scatterplots and extended it to Self-Organizing Maps (SOMs) including reliability encoding. I embed the automatic and visual comparison in an interactive analysis pipeline, which automatically calculates a set of representative descriptors out of a larger collection of descriptors. For a deeper analysis of the proposed result and the underlying characteristics of the input data, the analyst can follow each step of the pipeline. The approach is applied to a large set of chemical data in a high throughput screening analysis scenario.

For the analysis of time dependent, categorical data I propose a new approach called Time Parallel Sets (TIPS). It focuses on the analysis of group changes of objects in large datasets. Different automatic algorithms identify and select potentially interesting points in time for a detailed analysis. The user can interactively track groups or single objects, add or remove selected points in time or change parameters of the detection algorithms according to the analytical goal. The approach is applied to two scenarios: Emergency evacuation of buildings and tracking of mobile phone calls over long time periods.

Large time series can be compressed by transforming them into sequences of symbols whereas each symbol represents a set of similar subsequences in time. For these time sequences, I propose new visual-analytical tools, starting with an interactive, semi-automatic definition of symbol similarity. Based on this, the sequences are visualized using different linked views, each specialized on other analytical problems. As an example usecase, a financial dataset containing the risk estimations and return values of 60 companies over 500 days is analyzed.

Contents

3. Introduction

Modern technologies allow us to collect and store increasing amounts of data. As a manual inspection of these data overburdens the analyst, computer based approaches are increasingly used. Thereby, the applied methods depend on the analytical requirements and tasks which can be split in two groups. First, in some scenarios, clearly defined patterns should be found or hypotheses should be validated. In these cases, completely automatic algorithms can achieve excellent results as they can process large amounts of data very effectively. This contrasts to the exploratory data analysis, on which I focus in my work [Tuk77]. In this case, the analyst has no or only little knowledge about patterns or dependencies in the dataset. There may exist some a priori assumptions about the inner structure of the data, but their evaluation is not the only analytical target. Under these conditions, Thomas and Cook shaped a guiding principle: "Detect the expected and discover the unexpected" [TC05]. To discover these new and unexpected findings, the users have to be involved in the analytical process.

Visual Analytics enables the exploratory analysis of large datasets by combining data mining techniques and interactive visualization. Information visualization allows the presentation of complex data sets, so that connections and patterns can be discovered. But very large datasets quickly overburden the cognitive capabilities of humans. Therefore, visualization approaches often try to divide the data into different groups or level of details and, depending on the respective approach, show only one or some of these groups. Some examples of these strategies are "overview first, zoom and filter, then details-on-demand" [Shn96] or "Focus and Context" [CMS99]. Automatic data analysis helps to, e.g., preprocess the data or to identify potentially interesting structures in the dataset. Thereby, the analytical task is facilitated and the user can focus on integrating his special strength, the detection of previously unknown patterns and their evaluation, into the analytical process.

This process is not one-way, but rather a back and forth or looping [KKEM10] (see Figure 3.1). An examplary analysis process may run as follows: An initial preprocessing and visualization may allow the analyst to find some unexpected local patterns. Thereupon, the parameters of the data mining algorithms can be adapted and optimized leading to a clarification of the pattern detected. The knowledge obtained might be used to formulate hypothesis for the next round of analysis.

This idea of tight integration and combination of data mining, interactive visualization and expert knowledge is the basis of all Visual Analytics approaches. However, depending of the usecase and the structure of the data which should be analyzed, the particular methods and approaches may vary. Thus, different subfields within Visual Analytics emerged whereof I focus on two in this thesis: Visual Descriptor Space Comparison (Part I) and the Analysis of Time Series (Part II).

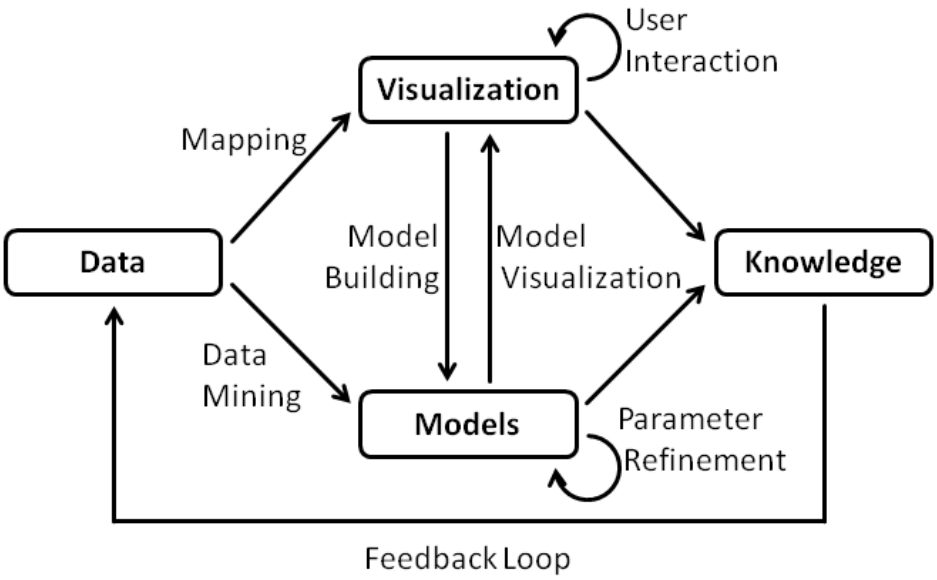


Figure 3.1.: This diagram shows the process of data analysis in Visual Analytics [KKEM10]. It tightly combines automatic data analysis, interactive visualization and the integration of user knowledge.

3.1. Contributions

My contributions to the two mentioned fields of Visual Analytics are the following:

- Comparison of hierarchically organized datasets:
  - I propose a novel comparison score reflecting structural differences and a visual-interactive comparison tool called ViPhy (see Section 4).
  - The tree comparison solution is extended by CloudTrees, a tree visualization specialized on the comparison of large trees with thousands of elements (see Section 5).
  - I introduce a color-based comparison technique for the visual comparison of data with unknown structure (see Section 6).
  - I present a interactive analysis pipeline to guide the user trough the analytical process (see Section 7).
- Analysis of time dependent data:
  - For the analysis of time dependent categorical data, I suggest TimeParallelSets (TIPS) combining a parallel coordinate-like visualization with novel algorithms for the automatic selection of key points in time (see Section 9).
  - I support the analysis of time sequences with an interactive combination of linked views, each specialized on specific analytical tasks (see Section 10).

Part I.

Visual Descriptor Space Comparison

The first part of this thesis focuses on the comparative analysis of descriptors. Often, the analysis of large amounts of data requires the support of computational methods e.g., to find patterns and outliers. These processes usually rely on a symbolic representation of each data-object. Complex objects like 3D models, graphs or chemical compounds exhibit multiple different aspects like size, curvature, charge, solubility, connectivity etc. which can be numerically described. Some aspects like the quantity of edges in a graph can be described with a single number, others need more values e.g., the three dimensional position in space and complex structure based definitions of molecules can include hundreds of values. In some areas, numerical descriptions are extended by categorical properties (e.g., countries of origin) or even more abstract symbolic representations like base- or aminoacid sequences in biology. Each of these representations is a so called **descriptor**, which can be used on its own or can be arbitrarily combined resulting in a new descriptor. The set of all descriptors is called **descriptor space** (see Figure 3.2).

The analytical tasks using descriptor representations can mainly be divided into two areas. First the general exploratory analysis, on which I will focus in this thesis and second the more specialized cases like search or classification. In the exploratory analysis, the user tries to gain knowledge about the underlying statistics and characteristics of a dataset. Usually, no prior knowledge or assumptions are integrated in the analytical process. The descriptors used should cover a broad variety of different properties of the data. Based on this, the interactive visualization should support the analyst e.g. in discovering connections between objects in the context of a particular descriptor or descriptor combination. In more specialized use cases, the descriptors are designed for a given task like clustering or retrieval. There exists no representation which is perfect for all requirements, but in general, the ideal descriptors should be as compact as possible while still including as much information as needed. Often, there exist no descriptor combination which is optimal for all objects in a scenario. An example in a retrieval setting is shown in Fig 3.3. A database of 3D objects was searched for the most similar objects for the input objects a human or a chair highlighted by the red border. The searches were performed on two different representations of the dataset (descriptor A & B). Descriptor A was able to capture aspects which offer a high discrimination for human models but the performance for a chair was quite poor. In contrast, descriptor B works better for the chair than for the human. In consequence, a proper description is heavily use case dependent, here, the user knowledge is very important for the evaluation of the results.

In the given exemplary search scenario, it is easy to asses the results and appropriate descriptor combinations can be trained based on a relatively small labeled training set. However, in most exploratory search scenarios, this is not the case. For example, in the exploratory analysis of high throughput screening (HTS), an overview over thousands of molecules is needed. If different representations of the whole dataset are presented to the user, their effective comparison is difficult (see Fig 3.4). An interesting aspect in one use case, e.g. the cost of the synthesis of a chemical compound, might be more important in a business environment than in research or more important developing a drug against cancer than again headache. As already mentioned, the descriptor representation is for example used to discover patterns in the data. But this does not mean that the descriptor which exhibits the strongest patterns is always the best. Often, strong patters are already known to the analyst and therefore in this context not interesting. This factor impedes a fully automatic descriptor comparison. In consequence, it is beneficial to include the user for evaluation in the descriptor analysis process. As the analysts are experts in their field (e.g., biology or finance), but often no experts in the data mining domain, it is important to allow for an easy access. In my approaches, I enable them to compare different descriptors and calculation results interactively, supporting e.g., the identification of similarities of differences between given descriptors.

In conclusion, the following tasks may arise, depending on the respective dataset and the analytical focus:

- Explorative comparison of different descriptor representations of a dataset, e.g. for the identification of differences and similarities.
- A similarity rating of the descriptors on a global and local level.
- A compact selection of descriptors, showing the main patterns of the dataset.

The particular approach to solve these tasks depends on the structure of the input data. The following chapters focus on the on the interactive comparison of hierarchically structured data where each descriptor forms a hierarchy (see Section 4 and 5) and on data, where no particular structure is assumed a priori (see Section 6). Although the particular techniques like the distance measurements or the visualizations are specialized on the respective input data, the overall procedures have many commonalities. Therefore, I propose a Visual descriptor Pipeline, integrating both approaches (see Section 7).

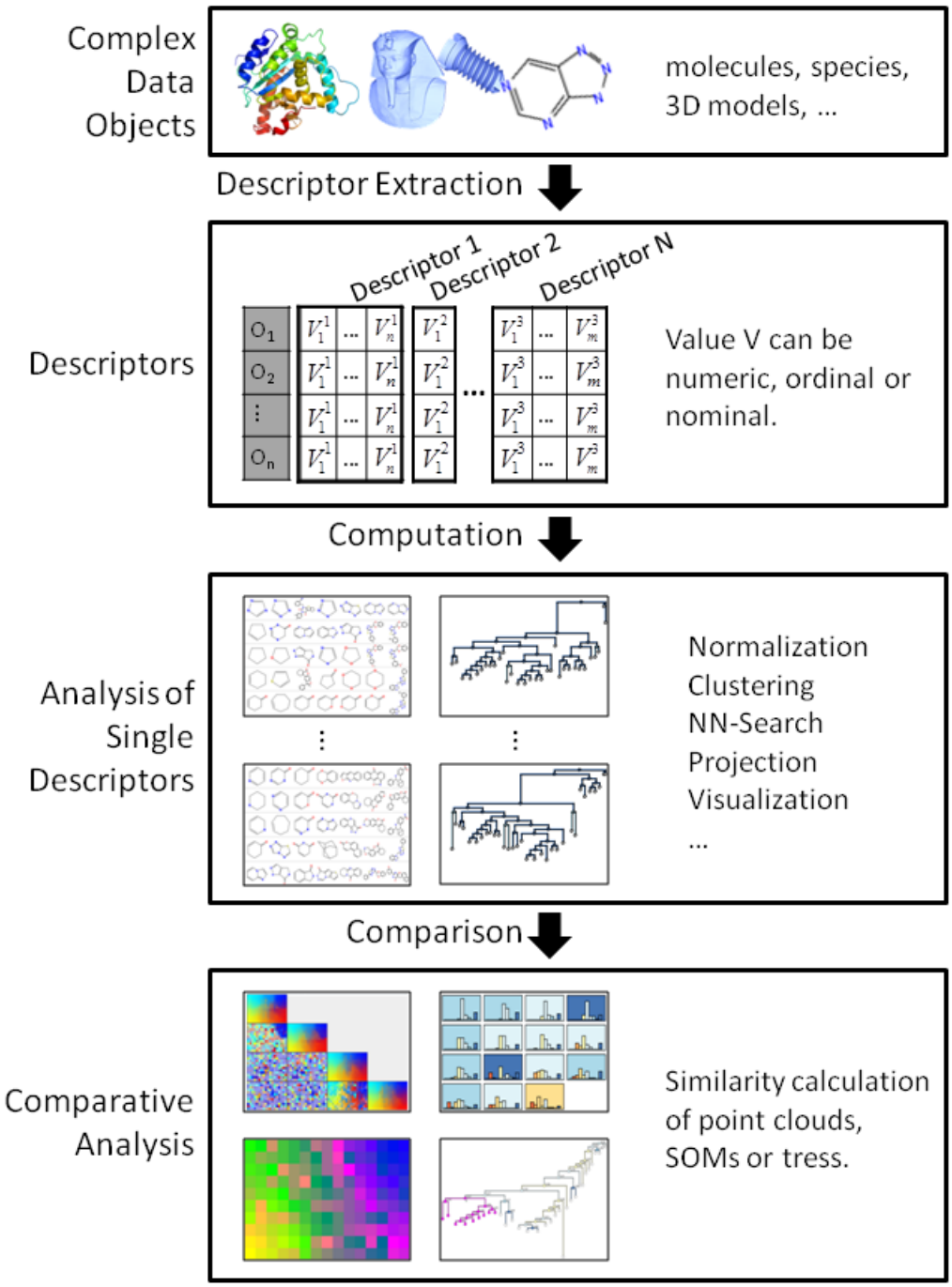


Figure 3.2.: The pipeline for computational data analysis. Descriptors, representing different aspects of complex data objects are extracted. Following, various computations based each of these descriptors can be calculated. The different results of these calculation are then compared to find differences and commonalities.

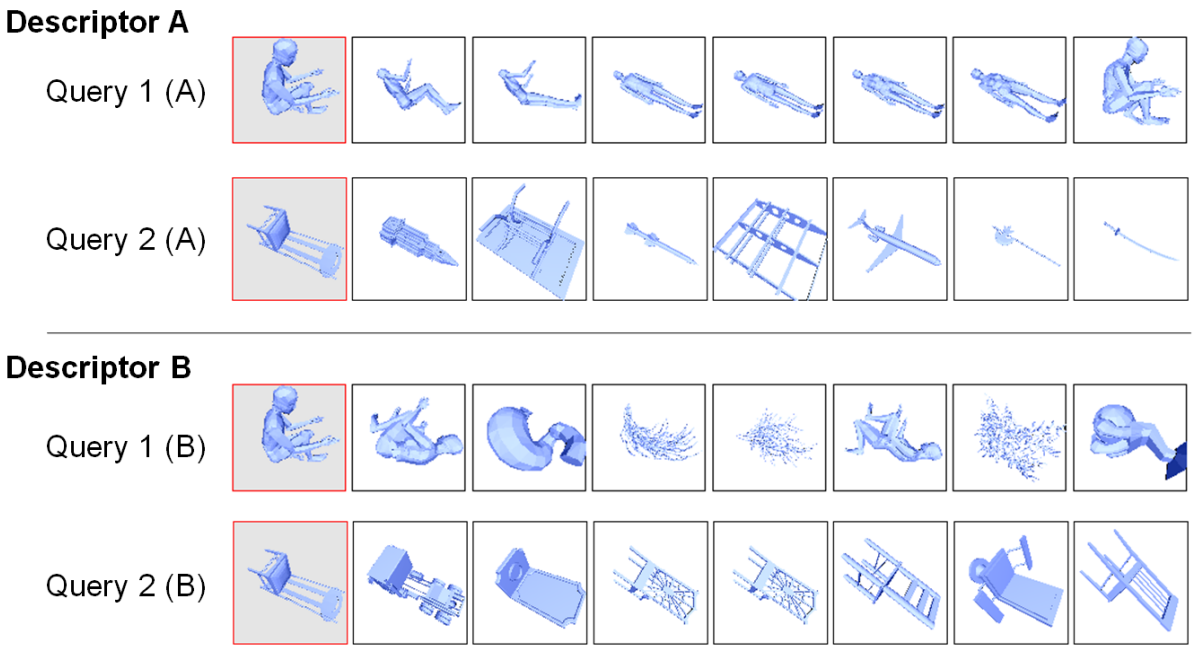
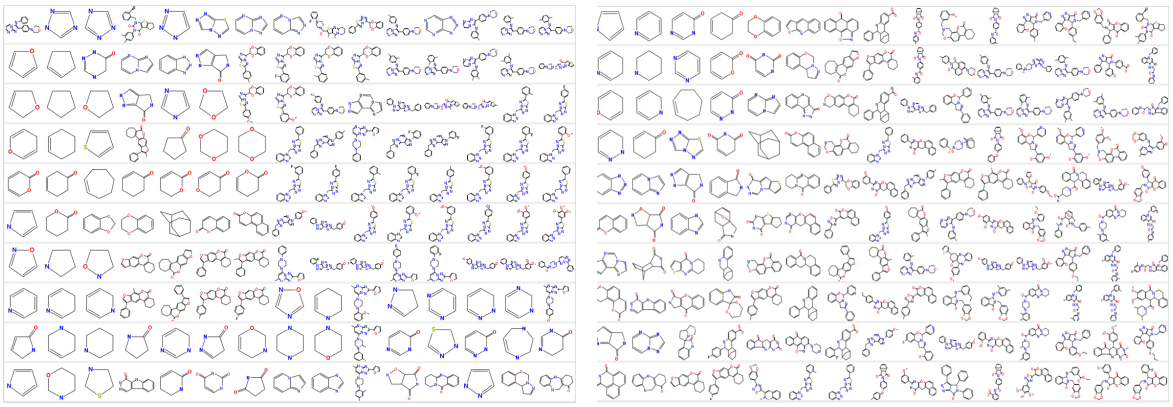


Figure 3.3.: This figure demonstrates the impact of different descriptors in a similarity search scenario in a 3D object data set. The first column, highlighted in red, represents the query objects, a human and a chair. The dataset is given in two representations by the descriptors A and B. Using descriptor A, human models, in contrast to the chair, can be retrieved very accurately. The object representation of descriptor B shows the opposite result. Hence, the choice of a representation is use case dependent.



(a) (b)

Figure 3.4.: This figure shows two different representations of the same chemical compound dataset. The self-organizing map projections have been calculated on basis of the number of nitrogens (a) and on molecule size (b). Both show comprehensible assembly, but to find differences and commonalities between them is difficult.

4. Visual Comparison of Datasets with Hierarchical Structure

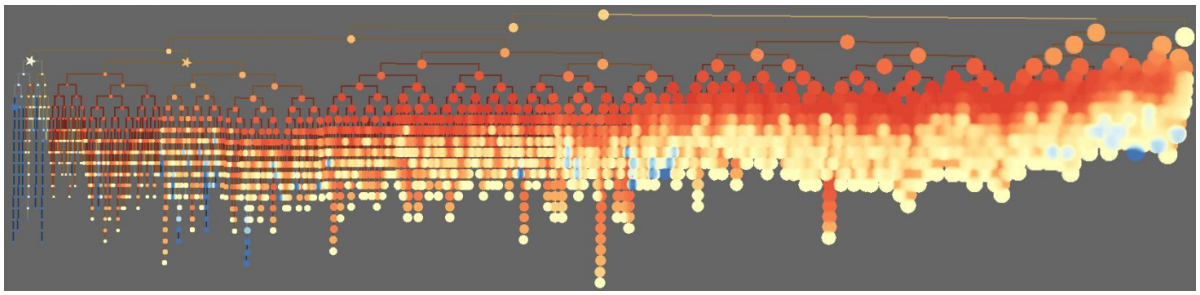


Figure 4.1.: Novel CloudTree visualization of a tree containing $>$ 7000 elements.

The comparison of multiple trees is an important issue in biology and other application areas, where each tree reflects the structure of the data based on one of the given descriptors. Either the underlying data is already hierarchically ordered, e.g. in the phylogenetic research or the hierarchic structure serves as tool for the analysis and is created e.g. using hierarchic clustering. In this thesis, I will give examples for both cases, starting with a hierarchically structured biological data (see Section 4.3) and later extending the approach to hierarchical clustered data in other application areas like pharmaceutical research in Section 5.

Biologists often analyze evolutionary relationships between organisms. Such relationships are typically represented by so-called phylogenetic trees. The derivation of such phylogenetic trees is usually based on a representation of each organism and an evolutionary model. There exist several representations of the same species e.g. different DNA- or protein sequences and varying assumptions on the evolutionary model which are reflected in algorithm and parameter choice. Thus, the derived trees are highly susceptible to parameter choices [LBX*01, Ham10]. Therefore, it is very important to compare sets of trees flexibly. The simultaneous analysis of multiple trees is expected to lead to more insight into the evolutionary processes and/or to compensate for uncertainties in the descriptor choice or model parameterizations.

In comparison setting, the analyst usually tries to find commonalities and differences among the compared entities. This is not restricted to global similarity evaluation, but more importantly, also encompasses the assessment of local patterns. For example, it is important to examine the stability of subhierarchies across the trees. The typical tasks include identification of globally interesting trees for reference purpose, finding locally dissimilar structures in trees with high global similarity to a reference tree, or the stability of a selected subhierarchy in other trees (see Section 4.3 for more information).

Currently, there are no readily available advanced methods for the visual comparison of multiple trees. The tools commonly applied by biologists support visualization of single trees as node link diagrams, e.g., using the FigTree [Ins] software. For multiple trees, a typical approach is a simple visualization of pairwise tree similarities

by a heatmap (see also Section 4.1.2). However, this approach does not satisfy the analytical needs, in particular, it does not provide structural tree comparison and assessment of local patterns.

In the analysis, we have to distinguish two different levels of complexity a) the number of trees to compare and b) the number of leaves in each tree. In the following, I will first focus on small trees to introduce my novel comparison score (see Section 4.2.2) and the interactive comparison system (see Section 4.2.4). An extension of the approach to large trees with thousands of elements is presented in following Section 5.
My main **contributions** are:

- I present a new visual analytics approach to compare multiple trees, both on global and local levels. To support efficient tree comparison, I combine automatic data analysis with interactive visualization (see Section 4.2.4). This combination allows for data analysis on several levels of detail. In particular, the results of automatic analysis are used for highlighting interesting patterns in the data and selecting data for detailed inspection.
- I introduce a new distance measure to compare rooted trees. This measure indicates differences in tree structure better than other available measures (see Section 4.2.2).
- I propose a normalization schema for the new element based score, which is applicable for other scores like the leaf based score, too. I allow the user a better assessment of the comparison results to prevent misinterpretation (see Section 4.2.3).
- I developed the CloudTree technique, extending the approach to enable the comparison of large trees with thousands of leaves (see Section 5).

My approach has various application benefits in phylogenetic (see Section 4.3) or pharmaceutical (see Section 5.3) research. I worked in close cooperation with domain experts from biology to compile a list of relevant tasks which is used to demonstrate the effectiveness my the approach to real world problems.

The work presented in this chapter is partially based on [BvLH*11].

4.1. Related Work

An overview of the visual analysis of graphs including their structural comparison is provided in a recent state-of-the-art report [vLKS*11]. It presents an exhaustive survey of work on tree visualization and analysis. Building on relevant approaches for interactive tree visualization discussed in Section 4.1.1, relevant work on visual comparison of multiple trees is recalled in Section 4.1.2. The visualization specialized on scalability issues for large trees will be discussed later in Section 5.1.

4.1.1. Visualization of Trees

The main approaches for visual tree analysis include node-link diagrams and treemaps. Node-link diagrams are well suited for the visualization of phylogenetic trees [HRR*07, LB06, DGB*08]. They allow for the representation of weighted edges and offer a very intuitive representation of binary trees as many users are familiar with them. The usage of links between nodes for larger graphs may be space inefficient. Therefore, specialized layout algorithms have been proposed to increase visualization scalability (see [vLKS*11] for an overview).

Alternative space efficient techniques, such as treemaps [Shn92a], use the whole available space. They recursively lay out child nodes within their respective parent nodes. As this technique employs overlapping of the parent nodes, users may find it difficult to assess the tree structure. This is a drawback especially for comparison

as the visually underrepresented inner nodes are sometimes more important than the leaves, which are similar in all trees.

To overcome space limitations and to support exploration of the tree structure, data analysis, visualization techniques and user interaction are combined. Several tree traversal and expansion techniques [EF10] can be employed to filter a given tree to the most interesting part. Alternatively, distortion techniques (e.g., fisheye views) allocate more display area to the parts of the tree of higher interest to the user. They can be based on a degree-of-interest function such as in DOITrees [CN02, HC04] or interactively selected by the user [MGT*03, TS08]. Multiple coordinated views [CMP09] offer an overview of the main tree structure and a detailed view on the selected parts of the tree. The construction of the overview relies on a score determining the interestingness of the substructure for a more detailed view.

These techniques focus on single trees and therefore form a basis for visual comparison of trees discussed in the following section.

4.1.2. Visual Tree Comparison

Existing techniques for visual comparison of trees focus on pairwise structural comparison and on comparison of multiple trees. An overview of existing visualization techniques and approaches can be found in [GK09]. A selection of approaches is laid out in the following section.

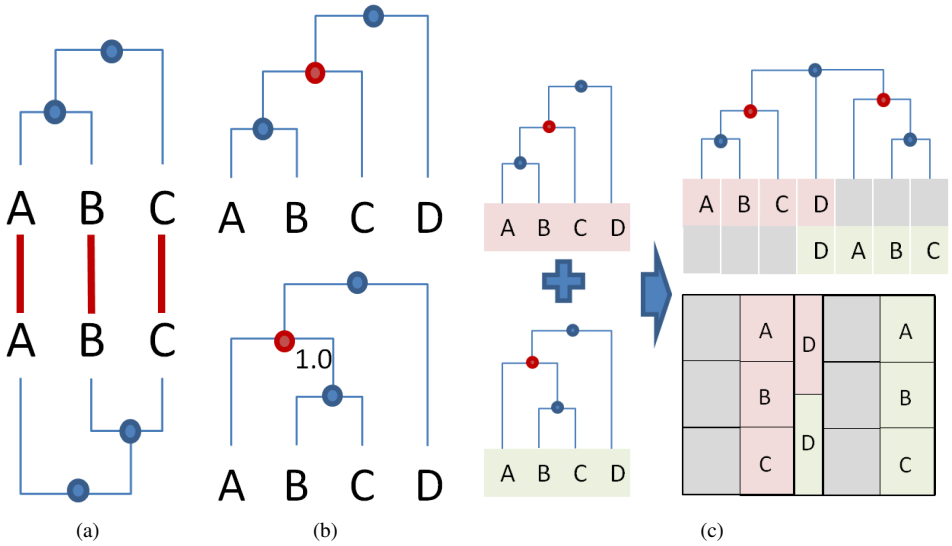


Figure 4.2.: Current pairwise visual tree comparison approaches.
a) Approach focusing on leaf node matching [HVW08]. b) Approach focusing on matching most similar structures [MGT*03]. The red nodes are the best matches based on a comparison score, yet different subtree structures exist. c) Tree comparison using union trees and contrast treemaps [TS07].

Pairwise comparison: An overview of general approaches is shown in Figure 4.2. Holten et al. [HVW08] proposed an approach matching leaf nodes between two trees. They draw the two trees in opposite parts of the

display and link their leaf nodes, while minimizing edge crossing. The crossing of links emphasizes the differences between the tree structures. These visual clues are enhanced by edge bundling. As shown in Figure 4.2(a), this approach does not reflect all structural differences between trees in some cases. In this case, the leaf nodes are fully aligned, however the tree structure differs. Telea and Auber extended this approach to analyze a sequence of pairwise tree comparisons [TA08].

Visual tree comparison focusing on the identification of corresponding nodes between the trees was presented by Munzner et al. in the TreeJuxtaposer system [MGT*03]. The approach was developed specifically for the analysis of phylogenetic trees. It analyzes and highlights leaf set similarities. This match is performed on demand, when the user clicks on a subtree, what can make it cumbersome to analyze large trees. The matching uses a similarity score based on set overlap of the leaf nodes (see Section 4.2.2.1). This score regards common groups of nodes without their structural relationship. As seen in Figure 4.2(b) two nodes (red color) are matched with the highest score, although their subtree structures differ.

Tu and Shen [TS07] propose a comparison of two trees in a treemap visualization called "contrast treemap". It was developed to support a static comparison of dynamic trees in two time points. It unifies the two trees to be compared for structural match. The visualization of the union tree employs specific node coloring and texturing which highlights value differences between leaf nodes. This approach is well suited for comparing value changes, however spotting structural differences above leaf node level is more difficult and is layout dependent. Moreover, the union tree algorithm leads to node duplication and thus to larger trees, which may complicate visual comparison (see Figure 4.2(c)). Koop et al. [KFS13] extended the approach to the merging of compared graphs in one hyper-graph but the resulting graph tend to become very large and complex if many compared dissimilar input graphs are compared.

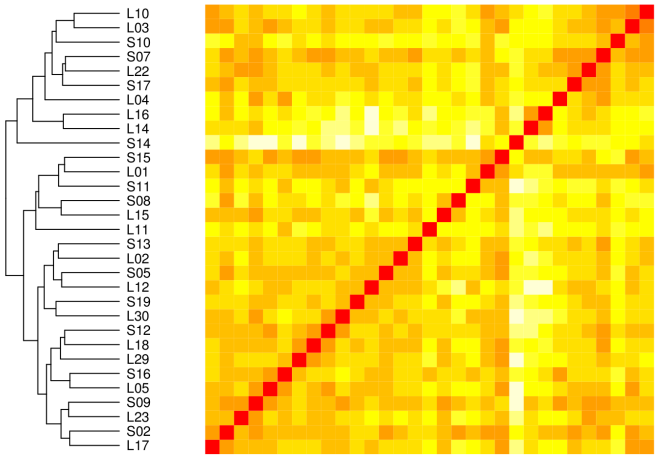


Figure 4.3.: Typical visualization of a multiple tree comparison in biology. Shown are pairwise distances between phylogenetic trees and their hierarchical clustering as computed by the TOPD/FMTS package [PGM07]. Such global analysis neglects any local structure or emergence of patterns in different trees. Note: S*x* and L*x* entries represent the name convention in microbiology for biomolecular sequences used to derive a particular tree.

Comparison of multiple trees: The above mentioned approaches are designed for the comparison of pairs of trees. There are only few techniques dealing with the comparison of multiple graphs (incl. trees). The avail-

able techniques for comparing multiple graphs [FPSG10, vLGS09] are not specialized on trees and do not allow for explicit visual comparison of tree structures. For trees, the so-called "Trees of Trees" approach constructs a meta-tree by successive joining of the underlying trees so that the total distance between tree nodes is minimized [Nye08]. This is a computationally intensive approach. The visualization of the result does not offer direct insights into the inner structural comparisons between the connected trees and does not provide for shared pattern identification. Normally, biologists use simple visualization of pairwise tree similarities in a heatmap [PHH10] combined with hierarchic clustering (see Figure 4.3). It also does not offer structural tree comparison and assessment of local pattern differences.

Hillies et al. presented an approach for the comparison of many trees [HHJ05]. Each tree is represented in a scatterplot using Multidimensional Scaling to determine its position. For a more detailed comparison, consensus trees are build revealing common substructures. Details about the uncommon structures are not provided.

4.2. Approach: ViPhy

My approach supports the comparison of multiple, rooted trees with identical leaf elements. It is designed to support identification of similarities and differences between these trees. This is not restricted to global similarity evaluation and supports the assessment of local patterns specifically.

Scalability of the analysis with respect to both the number of compared trees and their sizes (measured by the number of leaf nodes) is supported by visual analysis on several levels of detail. In this respect, we combine comparative data visualization and automatic data analysis. The computation of local and global similarities is used for filtering and highlighting interesting data patterns.

The analysis system, called ViPhy combines several interlinked views representing multiple levels of detail in the comparison analysis (see Figure 4.4). An initial overview shows the similarity matrix between all trees (see Section 4.2.4.1). From there, one reference tree can be selected for a detailed comparison with other trees (see Section 4.2.4.3 and 4.2.4.2). All views are supported by integrated calculation of local and global similarity measures (see Section 4.2.2).

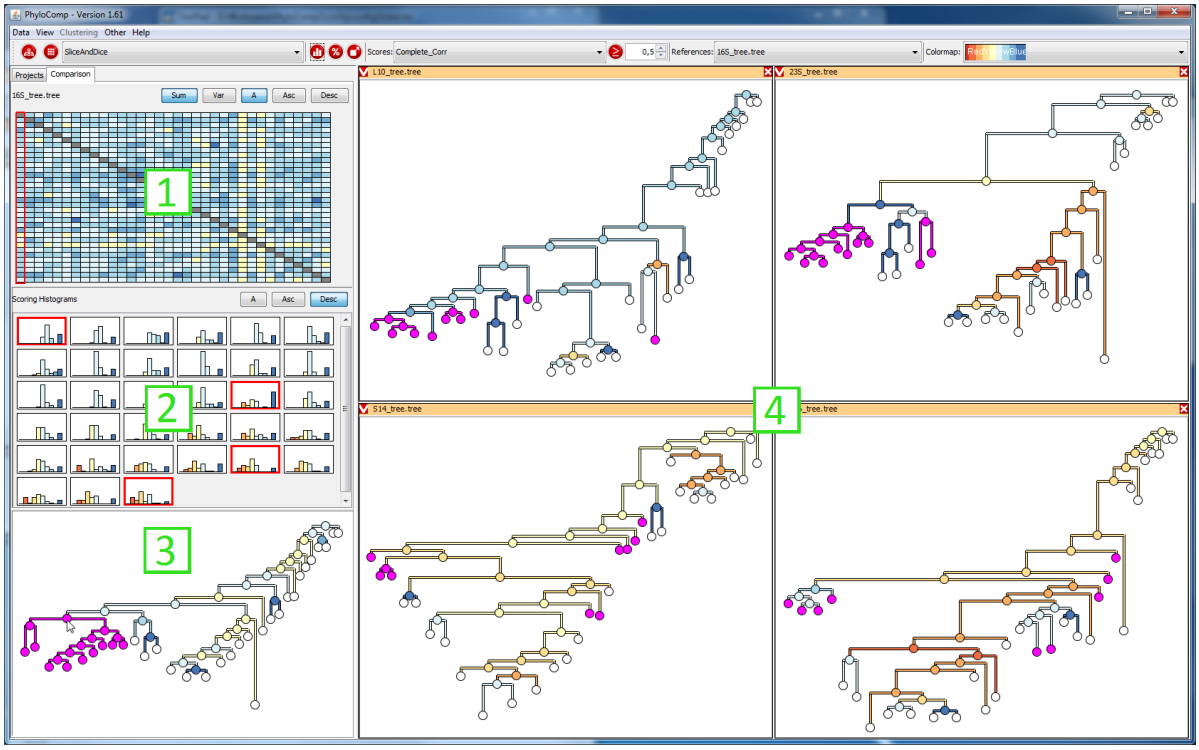


Figure 4.4.: Overview of our approach showing a visualization of multiple levels of detail for tree comparison. 1) Global pairwise tree distance matrix. 2) Score distribution for 1:*n* tree comparison. 3) Consensus tree for 1:*n* comparison with average matching scores. 4) Selected trees with scores showing similarity to the reference tree. Selected nodes (pink) are highlighted in all views.

In the following, the approach is described in more detail. The employed similarity measures are depicted followed by details on the interactive visualization.

4.2.1. Definitions

We are concerned with rooted trees exclusively in this study. A tree T consists of set of undirected edges E that connect pairs of nodes V, formally defined as $T = (V,E); E \subseteq [V^2]$. A tree T is called *rooted*, if one node r is distinguished as a so-called root node: $T = (V,E,r)$. A path in a tree is defined as a unique sequence of connected nodes $p(n_1,n_k) = n_1,n_2,\ldots,n_k$ where $n_i \in V$ and $(n_i,n_{i+1}) \in E$. A weighted tree has edges with associated real numbers as weights $w(e); e \in E$.

Note that in this work, we use the term hierarchy synonymously to rooted tree. *Leaf nodes (LN)* are nodes, that are connected only to one node. They do not have any child nodes. The set of all leaf nodes of a tree T is denoted as $L(T)$. Non-leaf nodes are referred to as *inner-nodes (IN)*. A binary rooted tree is a tree where each node apart from leaves has two child nodes (descendants). A *subtree* $T^n \subset T$ of a tree T is a tree consisting of a node $n \in T$ and all of its descendants in T. The subtree corresponding to the root node is the entire tree.

We denote the *distance* of two trees T_1 and T_2 as $d(T_1,T_2)$ with $0 \leq d(T_1,T_2) \leq 1$, so that small distances (close to 0.0) reflect high similarity of the two trees. The *similarity* is defined as $s(T_1,T_2) = 1.0 - d(T_1,T_2)$. The similarity of two nodes $n_1 \in T_1$ and $n_2 \in T_2$ is defined as the similarity of the subtrees $T_1^{n_1}$ and $T_2^{n_2}$ rooted at nodes n_1 and n_2, respectively. The *score* of the node n is defined as the maximum similarity to all nodes in the compared tree T_2.

The distance of two nodes n_1, n_2 in the same tree is defined as the length of the path connecting them. $d(n_1,n_2) = |p(n_1,n_2)| = |e_i|, e_i \in p(n_1,n_2)$. The weighted distance is defined as: $wd(n_1,n_2) = \sum w(e_i), e_i \in p(n_1,n_2)$.

We define *elements* of a tree as a set of all leaf sets $L(T^n)$ of all subtrees $T^n \subset T$:
$Elements(T) = \{\{L(T^n)\}, \forall n \in T\}$.

4.2.2. Similarity Measures

There are several approaches to calculate the distance between two trees [SP93, ZMP97, MGT*03]. I developed a novel element-based scoring scheme, reflecting differences in the inner structure of the compared trees. To support different analytical tasks, e.g., ignoring the structure and focusing only on leaves, two commonly used measures [SP93, MGT*03] are employed, additionally.

4.2.2.1. Leaf-Based Measure

Leaf-based scores calculates the similarity of trees T_1, T_2 based on their contained leaves $L(T_1)$ and $L(T_2)$. A normalized variant of the Robinson-Foulds distance is employed, following the strategy of Munzner and Guimbretiere introduced in [MGT*03]:

$$s(T_1,T_2) = \frac{|L(T_1) \cap L(T_2)|}{|L(T_1) \cup L(T_2)|}. \tag{4.1}$$

As only the leaves are included in the score calculation, the tree structure is ignored. Consequently, two (sub)trees containing the same leafs are classified as similar even if their structure may differ substantially. An example can be seen in Figure 4.5(a), where all roots have the maximal score. In this case, however, the internal structure of the trees differs significantly. Therefore, I developed a new measure presented in the following section.

4.2.2.2. Element-Based Measure

I present a new element-based score, which extends the leaf-based measure, so that it reflects the inner structure of the tree. In contrast to a leaf-based scoring, the inner-nodes of a tree are incorporated in the score.

$$s(T_1,T_2) = \frac{|Elements(T_1) \cap Elements(T_2)|}{|Elements(T_1) \cup Elements(T_2)|},\ where\ Elements(T_i) = \{\{L(T^n)\}, \forall n_i \in T_i\},\ for\ i \in \{1,2\} \quad (4.2)$$

As an example, the comparison score of the roots of the left (T_l) and center (T_c) tree in Figure 4.5(b) is shown.
$T_l : \{A,B,C,D,[A,B],[C,D]\}$; $T_c : \{A,B,C,D,[A,B],[A,B,C]\}$;
$s(T_l,T_c) = \frac{5}{7} \approx 0,71$

This score discriminates structural and node-based differences between trees more profoundly than the leaf-based score (see Figure 4.11). In many cases, as exemplarily shown in our use case, the score distribution is less skewed and the scores more homogeneously distributed across the value range. This allows for better discrimination of structural differences based on score values.
Moreover, as we include also inner-nodes in the calculation, no special score for the root nodes in trees containing the same organisms is needed (see Figures 4.10 and 4.5(b)).

4.2.2.3. Edge-Based Measure

Scores, which rely on nodes only (leaf-based and element-based scores), typically do not take the differences in the edge lengths (i.e. weights) into account. However, the edge length encodes important biological information such as evolutionary similarity of species. Therefore, we include a weighted edge-based score, which measures the difference between the sum of path lengths between all pairs of leaves in the compared trees (see Figure 4.6). This measure is inspired by the approach of Steel & Penny [SP93], who proposed a metric based on the difference of the number of edges connecting leaves.

$$s(T_1,T_2) = 1.0 - \frac{ed(T_1,T_2)}{maxwd(T_1,T_2)},\ where \quad (4.3)$$

$$ed(T_1,T_2) = \sqrt{\sum (wd(n_i^1,n_j^1) - wd(n_i^2,n_j^2))^2},\ and \quad (4.4)$$

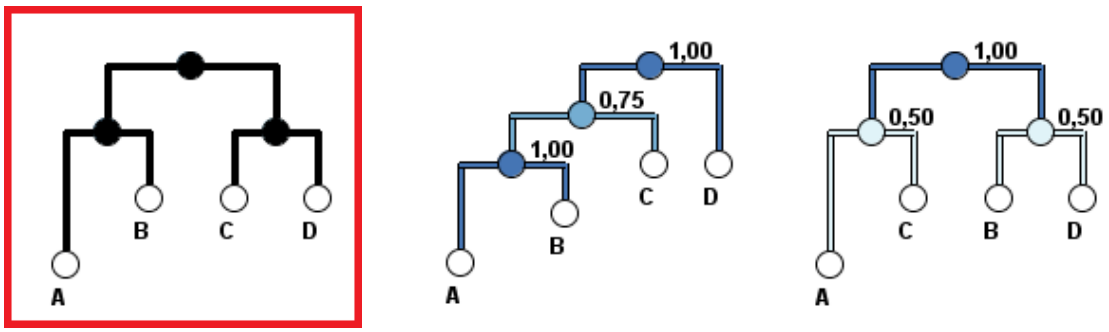
$$maxwd(T_1,T_2) = \max\{\max\{wd(n_i^1,n_j^1)\}, \max\{wd(n_i^2,n_j^2)\}\}, \quad (4.5)$$

$$\forall n_i^1, n_j^1 \in L(T_1), n_i^1 \neq n_j^1\ and\ \forall n_i^2, n_j^2 \in L(T_2), n_i^2 \neq n_j^2 \quad (4.6)$$

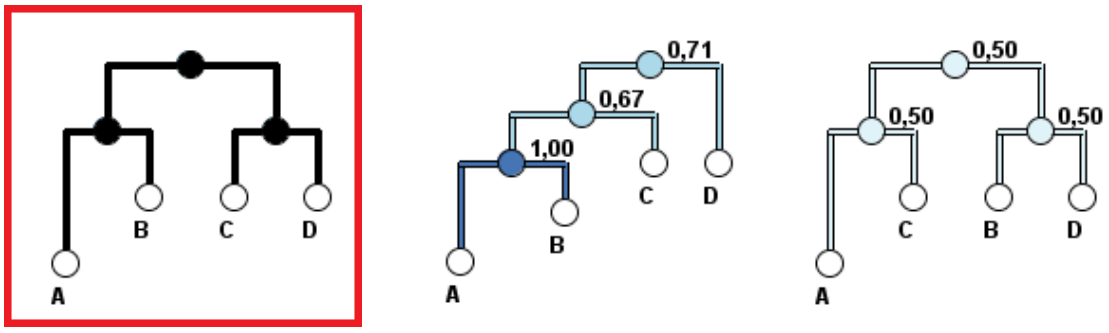
In this way, differences between trees containing the same elements in the same structure, but with other edge weight (i.e., length), can be revealed (see Figure 4.6). One disadvantage of the score is its normalization by the longest weighted path in both trees. As discussed in [SP93], this leads in large trees often to similarity scores close to 1.0 (see Figure 4.10(b)). Moreover, this score does not allow for determination of best matching nodes between two trees as only whole trees can be compared. Therefore, the score of a subtree is calculated according to the corresponding path lengths in the whole compared tree.

4.2.3. Score Normalization

The major problem of the element- and leaf-based scores [BvLH*11, MGT*03] is that they are not normalized. Depending on the sizes of the query and target tree, the similarity score does not fall below a certain thresh-

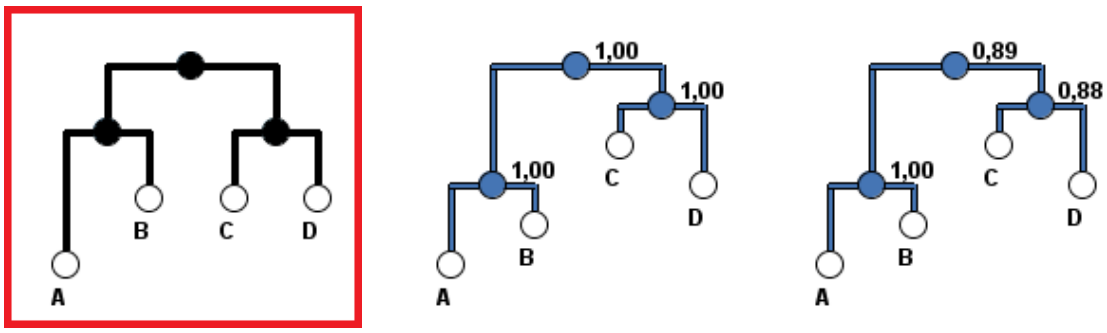


(a) Leaf-based score



(b) Element-based score

Figure 4.5.: This figure shows the comparison of two similarity measures. The scores calculates the similarity of a reference tree (left, with red border) to two other trees (center and right). The color map used ranges from blue(high similarity) to white (low similarity) (see Figure 4.9).
a) The scores are calculated using the leaf-based measure (Sec. 4.2.2.1).
b) The scores according to element-based measure. The score does not only include the leaves, it also considers the inner-nodes of the subtrees.



(a) Reference tree (b) Element-based score (c) Edge-based score

Figure 4.6.: A comparison of element-based and edge-based score for measuring similarity between a reference tree (left, red border) and another tree (center, right).
b) Element-based score. Due to equal elements, both trees are regarded as similar.
c) An edge-based scoring reveals the differences in edge length.

old, so it might be quite high although the match is very bad. For example, if two binary trees with the same leaves are compared using the element-based score, the result cannot be asymptotically worse than $1/3$ (see Section 4.2.3.1). So a seemingly high score of 0.4 does not reflect a good match. This might lead to misinterpretations, if the analyst is not familiar with the score calculation. As an illustration, we consider the tree similarity scores for hierarchically clustered real-world chemical components data obtained for different compound descriptors (see Figure 4.7 (a)). We see that the score (y-axis) increases with higher query tree size (x-axis). This is counter-intuitive as it is usually more difficult to find matching subtrees for larger query trees.

4.2.3.1. Normalized Similarity Score

I propose to normalize the score with respect to the theoretic maximum and minimum. The resulting normalized score gives information about the relative similarity of the compared trees.

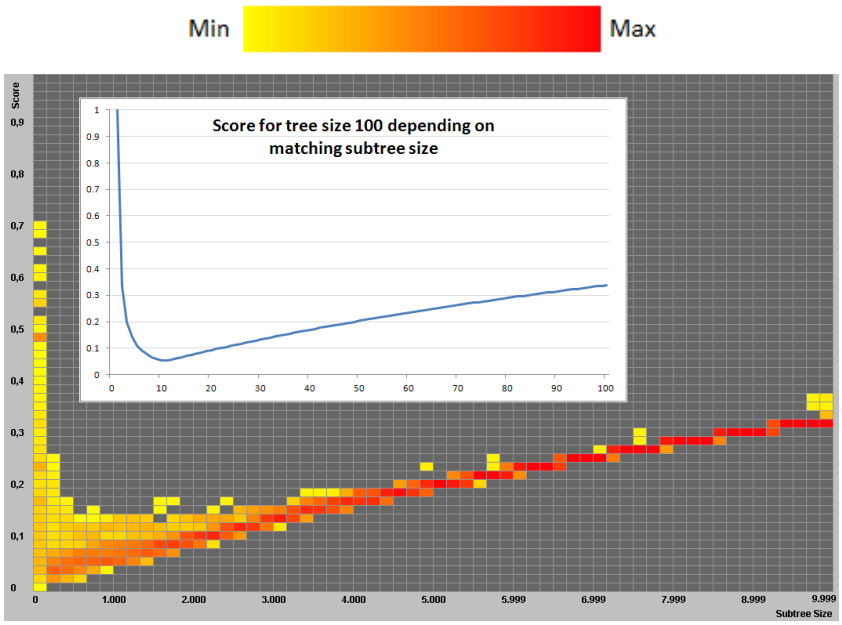
A linear normalization of the element-based score is employed (Eq. 4.7):

$$\widehat{s_e}(T_1,T_2) = \frac{s_e(T_1,T_2) - min(s_e)}{max(s_e) - min(s_e)} \tag{4.7}$$

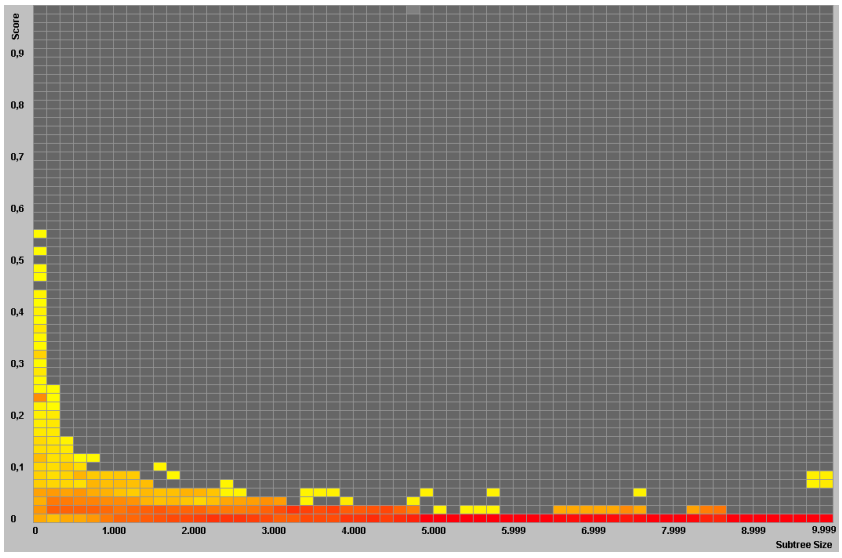
For the calculation, we need to identify the theoretical minimum and maximum score $(min(s_e), max(s_e))$.

The *maximum matching* score comparing two identical trees $T_1 = T_2$ is $max(s_e) = 1.0$.

The computation of the *minimum score* is not that trivial though. It is the score of the best matching unit (BMU) of query tree T_1 in the theoretically most dissimilar target tree T_2. We claim that the best matching unit for the worst case is either a single leaf node or the whole target tree. This depends on the size of the query and target tree T_1 and T_2. This relationship is non-linear as illustrated in Figure 4.7a.



(a) Score distribution without normalization



(b) Score distribution with normalization

Figure 4.7.: The need for score normalization exemplified by chemical structure comparison based on different descriptors. a) Scores without normalization. b) Scores with normalization based on theoretic score minimum values. The size of the subtree is mapped to the x-axis, its score to the y-axis. The color reflects the number of subtrees in one cell.

For the derivation of the minimum score, we first rewrite the element-based score calculation (see Section 4.2.2.2) using BMU notation (see Section 4.2.1): $M(T_1, T_2^m)$ with size M.

$$s_e(T_1, T_2) = \frac{M}{IN(T_1) + IN(T_2^m) - M}, T_2^m \in T_2 \tag{4.8}$$

Without loss of generality, we assume that $L(T_1) \subseteq L(T_2)$. This follows from the assumption of the same leaf nodes in all input trees and from the nature of subtree matching.

The minimum score for BMU being single leaf node:

$$min(s_e)^{Leaf} = \frac{1}{IN(T_1) - 1} \tag{4.9}$$

The minimum score for BMU being the whole tree T_2:

$$min(s_e)^{Tree} = \frac{LN(T_1)}{IN(T_1) + IN(T_2) - LN(T_1)} \tag{4.10}$$

The worst best score is then defined as the maximum of these two scores:

$$min(s_e) = \max\left(min(s_e)^{leaf}, min(s_e)^{Tree}\right) \tag{4.11}$$

The proof of this claim follows from minimizing the score (Eq. 4.8) with regard to all variables. As $IN(T_1) = E_1 = const$ in this case, it reduces to minimization with respect to two variables: $M(T_1, T_2^m) = M$ and $IN(T_2^m) = E_2$. This leads to a concave solution space with extrema at its boundaries, i.e., $M = 0$ and $E_1 + E_2 - M = \infty$. However, there are two constraints for the equation: 1) $M > 0$ (there is always at least one leaf that can be matched) and 2) $M \leq E_1$ as it is its subset. It then follows that the two extrema are achieved at boundary: $M = 1$ or $E_2 = IN(T_2)$. This means matching either one leaf or the whole target tree T_2 (see Figure 4.8).

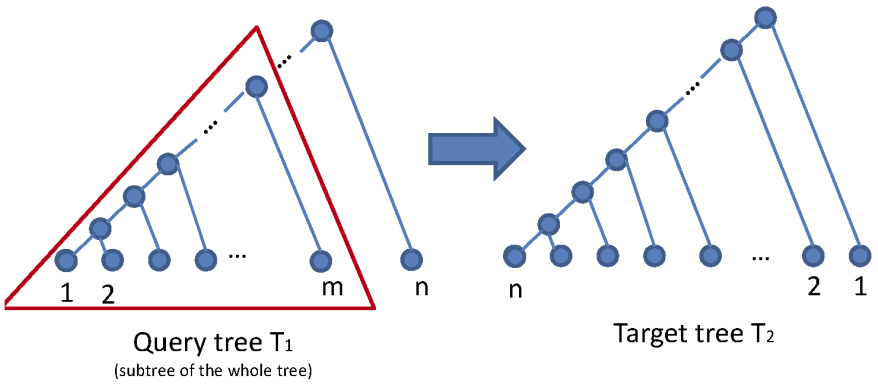


Figure 4.8.: Illustration of worst case matching of two trees. Both trees are stair-shaped with an inverted order of the leaves.

As an intuition of the result, assume that the best match is the whole target tree. Then the resulting score is:

$$s_e = \frac{LN(T_1)}{IN(T_1)+IN(T_2)-LN(T_1)} = \frac{a}{b} \tag{4.12}$$

If not all leaf nodes are matched, then the score changes as follows: $s_e^{-1} = \frac{a-1}{b-1}$, where $1 \leq a \leq b$.
If this results in a lower score, it is disadvantageous to reduce the size of the matched tree (i.e., the BMU is no longer the whole tree). If, in contrast, the score increases, the target tree can incrementally be reduced to its minimum, i.e., one leaf.

The border between the two types of match is the point where the scores for full tree match or leaf match are equal (i.e. $min(s_e)^{leaf} = min(s_e)^{Tree}$).
Then, for this border holds:

$$IN(T_2) = IN(T_1) \cdot (LN(T_1) - 1) \tag{4.13}$$

As an example, in the binary case, the border can be calculated using only the leaf count of the query and target tree:

$$LN(T_2) = \left\lceil \frac{1+\sqrt{LN(T_2)-1}}{2} \right\rceil . \tag{4.14}$$

□

4.2.3.2. Multiple Tree Comparison Score

For the comparison of multiple trees, we need to define a score, which enables us to compare a tree T with a set of trees $T^S = \{T_i\}, i = 1,2,\cdots,n$. We calculate such a score as an average of pairwise comparison scores.

$$\widehat{as_e}(T,T^S) = \frac{\sum_{T_i \in T^S} \widehat{s_e}(T,T_i)}{|T^S|} \tag{4.15}$$

This is called the *consensus score*. It provides information on the overall similarity of a tree to a set of compared trees. It can be calculated for all trees in the dataset or only for a subset. Alternatively, for better discrimination of scores, an average of the squared scores can be employed.

4.2.3.3. Summary

The similarity scores described above capture different tree properties. The proposed element-based score reflects the tree structure and provides a good score distribution. Leaf-based score is useful only in particular cases when global grouping of the leaves is of interest, solely. The edge-based measure captures a signatures of the general structure, and can accommodate edge weights. Nevertheless, its normalization and thereby the scalability to larger trees is an open problem.

When using efficient implementation, in optimal case, all scores exhibit the same computational complexity of $O(|V|^2)$ [SP93, MGT*03, LH03] which is optimal for a pairwise comparison of all subtrees [LH03].

4.2.4. Interactive Visualization

To ensure scalability, several visualizations representing different levels of abstraction are integrated in the tree comparison system (see Figure 4.4). It thereby enables the user to compare whole trees, subtrees or individual nodes. The visualizations are linked for interactive highlighting of interesting tree parts in all views. The applied calculation methods and visual attributes (e.g., colormaps) are consistent in all visual representations and can be chosen interactively. The tree visualization technique was chosen so that it best fits the demands of the comparison of phylogenetic or hierarchically clustered data. I employ the commonly used node-link technique with traditional, orthogonal top-down layout, because it is very beneficial for the user's perception [BKH*11], its easy interpretability (see Section 4.1.1) and the possibility to reflect the edge weights created in the tree building process [DGB*08].

In the following, the data views are presented in detail. We use the color map shown in Figure 4.9, where red represents low and blue high similarity. After testing various color maps from www.colorbrewer.org, this color map is used because it highlights high and low scores well and offers a good score discrimination.

Figure 4.9.: Color map used for the tree comparison. Colors based on www.ColorBrewer.org, by Cynthia A. Brewer, Penn State. [Bre12]

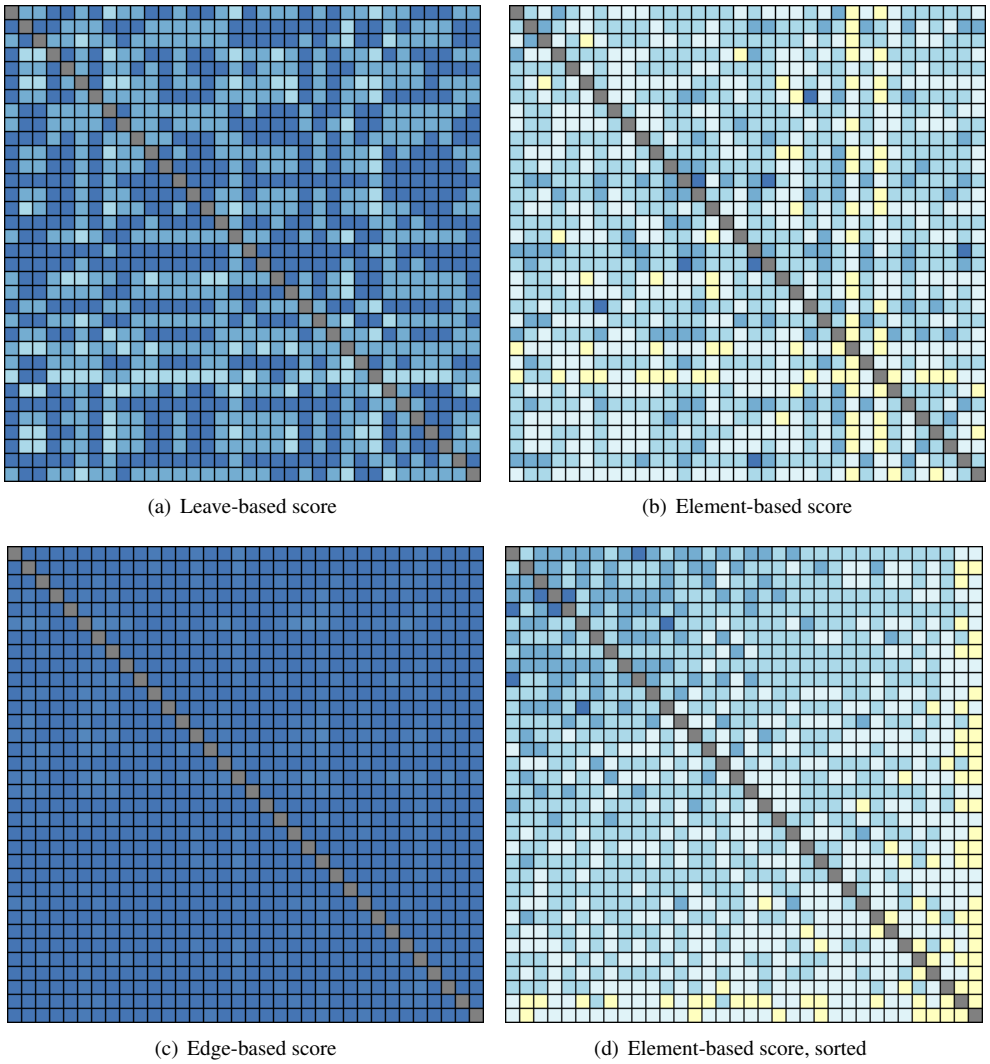
4.2.4.1. Comparison Overview

An overview of pairwise similarities of all trees in the test data set is presented by the similarity matrix. This representation is included as it is an established approach for visualizing pairwise relations, presenting a familiar and easy to grasp overview (see Figure 4.3). Every row/column represents one tree and the cells encode the global similarity evaluation of tree pairs. This view allows for an overview of global similarities among trees and thereby, offers the possibility to select a reference tree for more detailed 1:*n* tree comparison in other views.

Additionally, the matrix can be sorted according to the sum of the scores (Figure 4.10). In the sorted map, the most (dis)similar trees to other trees stand out on the sides of the matrix (e.g., two right columns represent most dissimilar trees to all others). To improve the matrix visualization, the variance of the scores is also included to indicate possible local differences. The variance matrix is shown alternatively to the score matrix.

4.2.4.2. Score Distribution View

The overview matrix offers only one global score per tree comparison. Therefore, a more detailed histogram view on the score distribution in the tree comparison is included. The histograms offer a compact overview of the score distribution of all nodes in each tree when compared with the selected reference tree (see Figure 4.11). These views allow the detection of distinct score distribution patterns such as predominantly high scores, bimodal score distributions (many low and many high scores), or trees with high variance of scores.
The histograms also serve for comparison of various score measures for the analyzed data set (see Section 4.2.2). Figure 4.11 shows a comparison of the score distributions for the three scores used. The element-based score shows better discriminative power for tree comparison than the other two measures. For example, the score



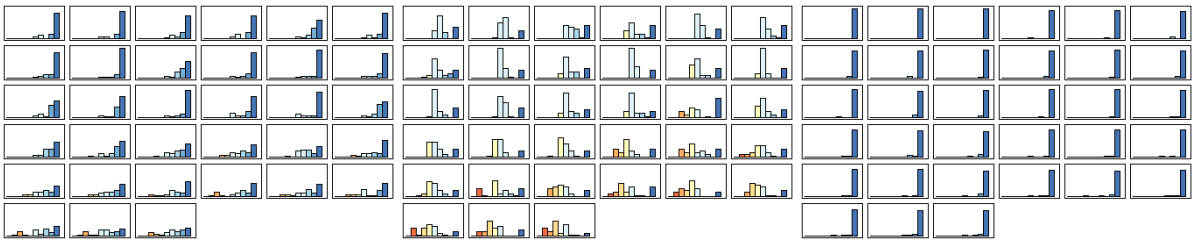
(a) Leave-based score (b) Element-based score

(c) Edge-based score (d) Element-based score, sorted

Figure 4.10.: The similarity matrix color codes the pairwise global similarity of the compared trees based on a selected score. a) Leaf-based score b) Edge-based score c) Element-based score d) Element-based score with descending matrix sorting. The element-based score shows the best discrimination of trees in contrast to the edge based score which barely reflects differences among trees.

distribution of the leaf-based measure is strongly skewed towards high scores. In contrast, the new element-based measure has more widely distributed scores.

The histogram view is also used for selecting a subset of trees for a detailed pairwise comparison with the reference tree (see Section 4.2.4.4).



(a) Leaf-based score (b) Element-based score (c) Edge-based score

Figure 4.11.: This figure shows the histograms of the scores of the elements of all trees compared with one reference tree, in descending order. The scores are calculated using (a) the leaf-based, (b) the element-based, and (c) and the edge-based score. As also can be seen in the overview matrices (see Fig 4.10), the element-based score shows a better discrimination than the other two measures.

4.2.4.3. Consensus Tree View

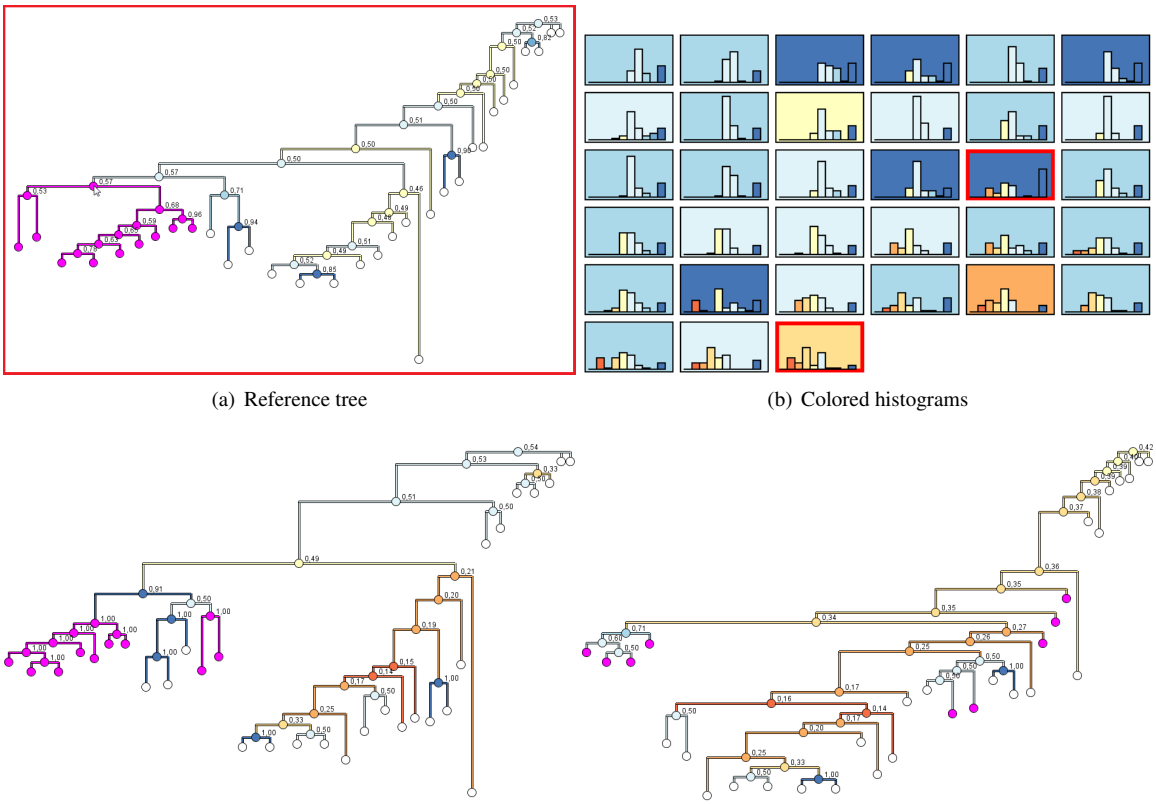
The consensus tree provides a compact form of a 1:*n* comparison between one reference tree and all other trees (see Figure 4.12a). This view shows the conservation of the nodes of the reference tree throughout the data set. Each score is the average of the scores comparing a reference tree node against its best matching unit in all other trees. High scores stand for good matches across the data set. In this way, the biologist can immediately see, which sub-hierarchies are conserved (see Figure 4.12a).

To gain more details on the composition of the calculated score of a selected node, its distribution in the data set needs to be analyzed. Therefore, an interactive functionality is included, which allows the user to highlight their values in the data set on demand. In particular, the backgrounds of all histograms of the compared trees are color coded according to their similarity to a selected node in the reference tree (see Figure 4.12b)). This is a powerful tool to quickly detect trees with either similar or dissimilar sub-hierarchies.

4.2.4.4. Tree Comparison View

The tree comparison view is the most detailed view in the framework. It contains the comparison tree scored against the reference tree shown in the consensus tree. I extended the well known pairwise comparison view with interactive functions for a better analysis of the data (see Figure 4.4 (3) and (4)). The scoring scheme can be chosen from the proposed set of measures (see Section 4.2.2).

The linked views of tree comparisons allow for highlighting of selected structures and their best matches in all visible trees. This feature has been proven useful for comparing trees [MGT*03]. In an exemplary analysis setting (comparison of phylogenetic trees), it reveals the particular distribution of the organisms. Thereby, it provides additional information on tree similarity (see Figure 4.12). The figure shows a reference tree with a highlighted subtree rooted in a user-selected node (pink color). The best matching nodes of the subtree are



(a) Reference tree

(b) Colored histograms

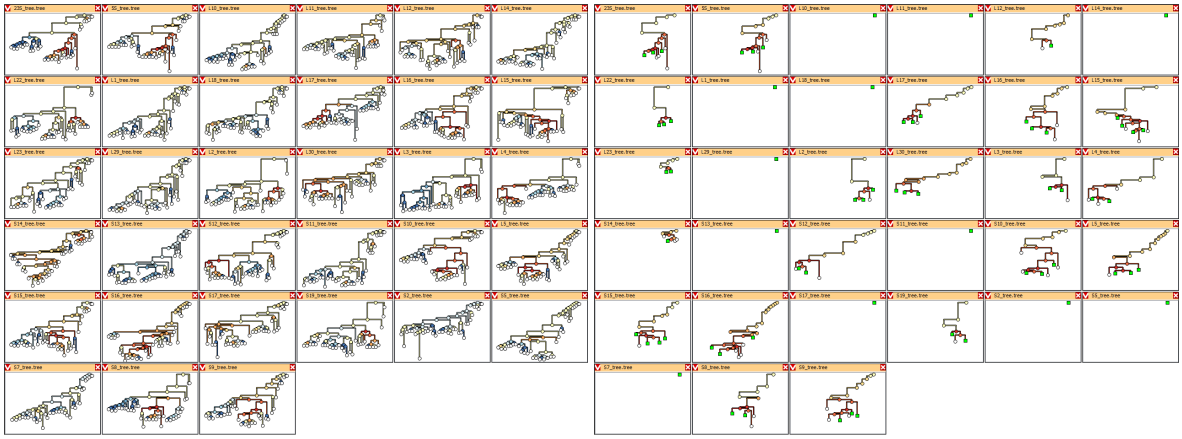
(c) Compared similar tree

(d) Compared dissimilar tree

Figure 4.12.: Tree comparison view with highlighting. a) The reference tree. One of its subtrees is highlighted by the user (in pink). b) Histogram backgrounds are colored according to selected node in reference tree. Trees selected for a detailed analysis (c&d) are highlighted with a red border. c) The compared overall most similar tree. d) The compared most dissimilar tree. c) & d) The leaves of the subtree highlighted in the reference tree are highlighted in the compared trees. Highlighted elements are more diversified in the dissimilar tree and close in the similar tree.

highlighted in the two compared trees. As expected, the highlighted elements are much more distributed in the dissimilar tree and compact in the similar tree.

To reduce the complexity of the visualization and improve scalability, the tree representation can be simplified [CMP09]. In this case, those subtrees are collapsed that have element scores below a user-defined threshold (see Figure 4.13, green rectangles). By this, structures similar to the reference tree are hidden and dissimilarities are pointed out. In analogy, all subtrees above a certain threshold can be collapsed. This allows for focused analysis of similarities among trees. Additionally, branches that do not lead to collapsed nodes can be hidden. The threshold slider allows for interactive setting of the threshold and exploration of the effects of threshold change on the tree collapsing.



(a) Original tree view (b) Collapsed tree view

Figure 4.13.: Tree comparison view with collapsing of similar elements. a) The original view with all nodes visible. b) The collapsed view, where all elements below a user-defined threshold are collapsed for focused view on tree dissimilarities and better scalability of the view. The green rectangles represent the collapsed nodes.

4.3. Application to Ribosomal Phylogenies

The application section provides an insight into the usage of the proposed approach for visual comparison of multiple phylogenetic trees in current biologic research activities. The use case has been provided by biologists with whom I cooperated (Dr. P. Weil and Prof. Dr. K. Hamacher, Institute for Computational Biology & Simulation, TU Darmstadt).

Phylogenetic trees represent the evolutionary relationships between organisms. These relationships are usually calculated based on alignments prepared from biological data such as DNA- or protein sequences. The leaves correspond to the respective organisms, while the branches denote the evolutionary ancestry between them and each inner node represents an (unknown) ancestor.

The derivation of such phylogenetic trees is usually based on a representation of each organism and an evolutionary model. There exist several representations of the same species e.g. different protein sequences and

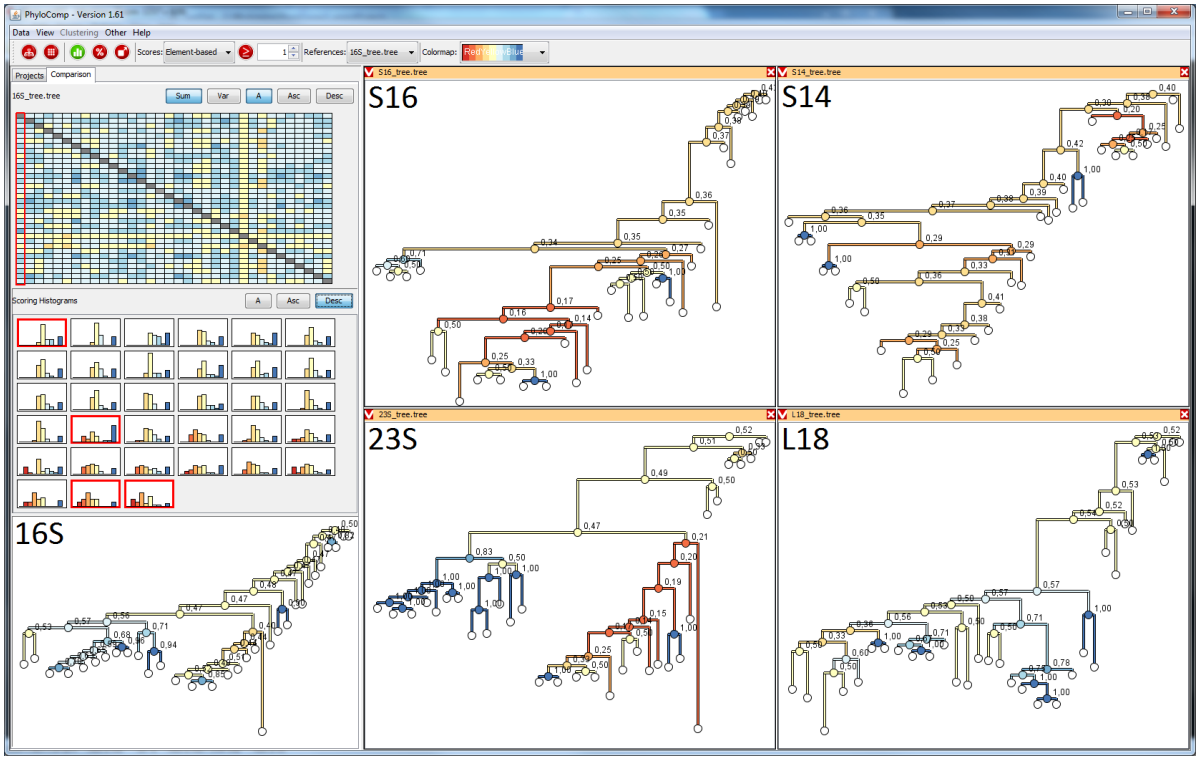


Figure 4.14.: User interface in the application using 34 phylogenetic trees with 32 organisms each. The 16*S* tree was chosen as a reference. Four trees (*L*18, *S*14, *S*16 and 23*S*) were chosen for detailed comparison. The selected trees are highlighted in the histogram view with red border.

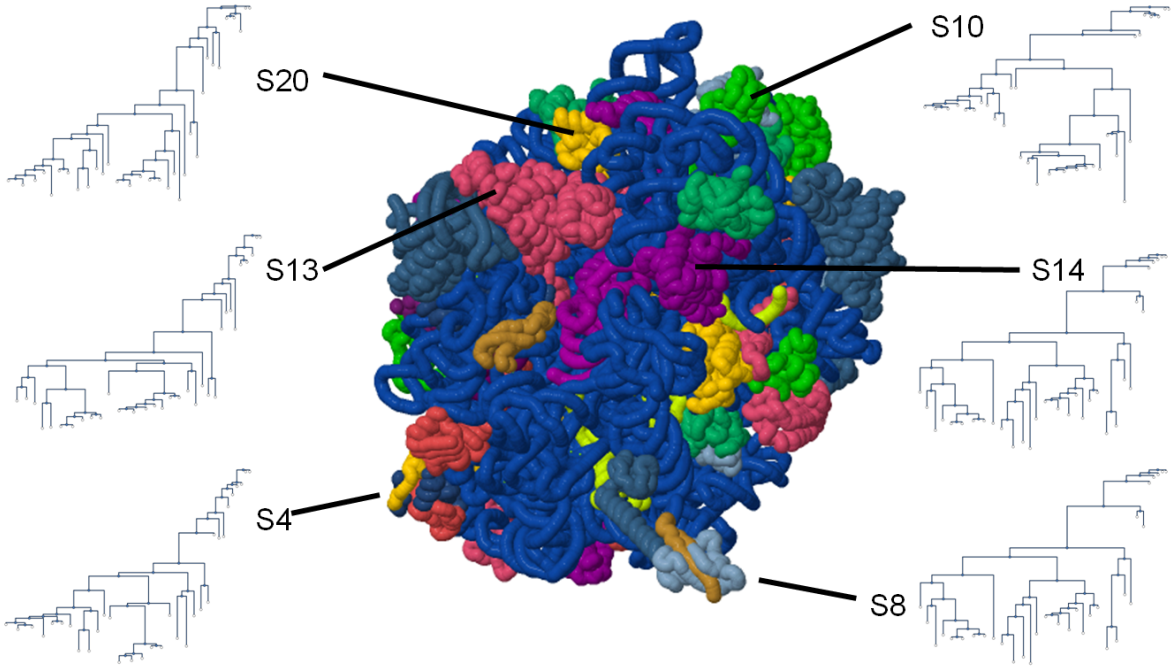


Figure 4.15.: Subunits of the ribosome (center) are often used as basis to calculate phylogenetic trees. Different subunits lead to different trees.

varying assumptions on the evolutionary model, which are reflected in algorithm and parameter choice (see Figure 4.15). Some of the parameters like the substitution matrix used for sequence alignment of proteins can only be estimated as it represents the evolutionary distance which itself is the subject of examination. Thus, the derived trees are highly susceptible to parameter choices [LBX*01, Ham10]. The simultaneous analysis of multiple trees is expected to lead to more insight into the evolutionary processes and/or to compensate for uncertainties in the descriptor choice or model parameterizations.

A typical research project deals with 10 to 50 phylogenetic trees or even more, while the number of organisms in the analysis spans orders of magnitude, from highly specialized questions on some ten, up to thousands of entities obtained from high-throughput analysis protocols. If we focus on phylogenetic trees, we assume $n \geq 2$ rooted trees with the same leaf elements (organisms). Although the trees presented in this application are predominantly binary, the approach is not restricted to them.

Biologists focus on the analysis of evolutionary relationships among a selection of bacteria. The relationship between the selected bacteria is commonly determined by comparison of the 16*S* rRNA sequences. This comparison leads to the phylogeny, approved by the community of biologists. Nevertheless, such sequence comparison can also be conducted using other sources such as conserved proteins, acknowledged to aid in finding the correct phylogeny. The ribosome is a molecular structure ubiquitously distributed in all living organisms. To this respect, the biologists are interested in the analysis of the effect of the data source on the resulting phylogenies.

Thereby, they solve the following **tasks**:

T1 Identification of *globally interesting* trees for reference purposes and determination of various *patterns in subhierarchies* of the trees:

T2 The conservation of every subhierarchy of the reference tree in all other trees.

T3 Distribution of a selected subhierarchy throughout the dataset.

T4 Trees with high global similarity but low local similarity and vice versa.

T5 All (dis)similar structures in all compared trees with respect to the reference.

The comparison of phylogenetic trees is a starting point for further biological analysis. As this analysis is very time and resource consuming, the biologists need to concentrate on the most promising starting points. The presented tree comparison tool aids biologists to track down these highly promising pointers even in large data sets and to form working hypotheses for research.

4.3.1. Phylogeny Calculation

The relationship of two organisms is calculated on a common data source. Choosing an appropriate data source for phylogenetic tree calculation is however still an area of ongoing research. In this use case, sequences encodings of different parts of the ribosome, namely rRNA and proteins have been employed.

The input data consists of bacterial genomic data downloaded from GenBank data base (GbDB). The biologists extracted the desired sequences encoding ribosomal parts out of the whole genomes using BioPython [CAC*09] and Hidden-Markov-Models (HMM) in the PFAM data base [FTM*07]. To calculate the phylogenetic trees, they applied the following web based tools of phylogeny.fr [DGB*08]. The alignments were performed using the MUSCLE [Edg04] algorithm. After the so-called "curation" of the alignments with GBLOCKS [Cas00] phylogenetic trees have been computed with PhyML [GG03, AG06].

This resulted in 34 phylogenetic trees with 32 organisms each. The trees are named according to the ribosomal parts encoded by source sequence (e.g, 16*S*, *L*4, ...). The employed naming convention is: letters before numbers for proteins and letters after numbers for rRNA. To confirm the suitability of the data sources used, the biologists compared the phylogeny of our calculated 16*S* data to a larger data set already validated by the community biologists [WHM*09]. The subset is in very good agreement with the one acknowledged by the experts. Thus the 16*S* phylogenetic tree can be used for the comparative study described in this section.

4.3.2. Results from Visual Analysis Process

The presented analysis process started with the selection of a reference tree based on the global similarity of the trees **(T1)**. The comparison of the implemented scores in the overview matrix showed that the element-based score exhibited the best discrimination properties (see Figure 4.10 and 4.11). Therefore, it was used in the following analysis. Potentially interesting reference trees are usually characterized by high or low scores compared to all other trees. These trees were identified in the sorted matrix (see Figure 4.10 (d)). Following this track, one would choose the *S*16 tree, because it showed up as the most distant tree. However, in this study, we concentrated on tree 16*S* as it refers to a biologically-validated phylogeny [WHM*09]. Looking at the global scores in the matrix column corresponding to this reference tree (see Figure 4.10 (b)), the biologist found out that the derived phylogenetic tree strongly depends on the underlying data set.

The sorted histograms below the matrix allowed the analysts to identify interesting patterns in the score distribution for the elements of each compared tree **(T4)**. Initially, their focus was on the most dissimilar trees. They identified the *S*14 and *S*16 ribosomal proteins as two of the most deviating trees to the reference tree (Figure 4.16

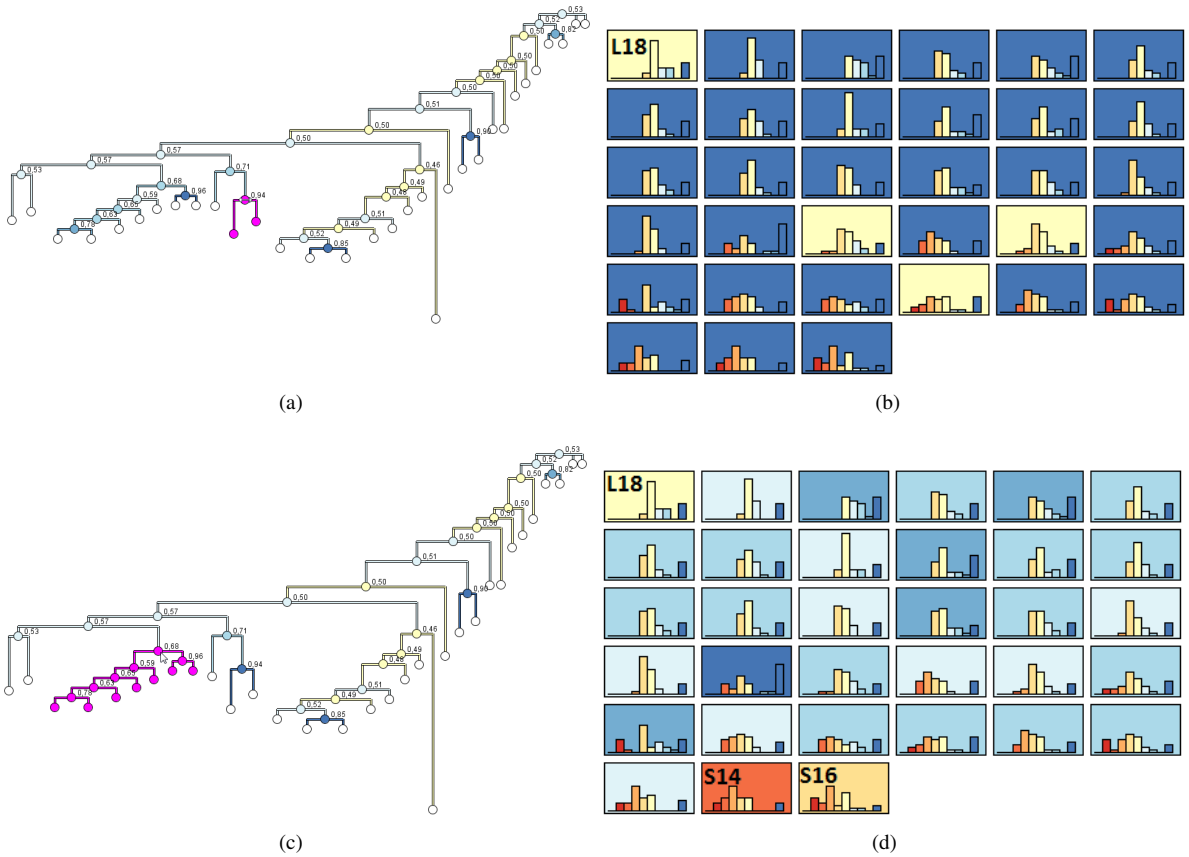
(d)). Moreover, these trees are most dissimilar to all other trees as well (the two right columns in Figure 4.10 (d)). This was an interesting finding, because the *S*14 protein is involved in the assembly of the ribosome suggesting a highly conserved function and thus a highly conserved sequence [HBB*08]. Additionally, the *S*14 tree exhibited a bi-modal score distribution **(T3)**. The distribution was characterized by a large amount of low- and high-scored elements (Figure 4.16 (d)). It therefore indicated similar and different structures to the reference tree inside the same tree.

The consensus tree offers the possibility to analyze the stability of subtrees across the dataset **(T2)**. Examination of the consensus tree identified two interesting, highly conserved clusters. Such a finding raised attention, because it contradicts the initial statement that "phylogeny computations strongly depend on the underlying data set". The closer analysis of the distribution of the two selected clusters **(T3)** in the trees revealed outliers with a low conservation. Besides the already conspicuous *S*14 and *S*16, the *L*18 tree attracted attention (Figure 4.16 (a) & (b)). Despite its high global similarity to the reference, it had a very low conservation for both selected substructures (Figure 4.16).

The identification of extrema throughout the whole data set **(T5)** can only be seen if all trees are simultaneously displayed. For a large number of trees, this leads to a cluttered view (see Figure 4.13 (a)). To reduce the complexity of the visualization, the biologist hid all substructures above a similarity of 0.25. This clearly reveals the most dissimilar structures compared to the reference tree (see Figure 4.13 (b)). Their closer analysis showed that *Xanthomonas campestris*, a plant pathogen, was present in the opened leaves of most trees. This indicated that the relative position of *Xanthomonas campestris* differed significantly from the approved 16*S* reference tree throughout the dataset.

This study helped biologists to take a closer look on function and structure of the ribosomal proteins *S*14, *S*16 and *L*18 for the identified species. They now especially concentrate on the question, why the phylogenies resulting from those protein sequences exhibited such organism clusters. This research could lead to previously unknown connections between the analyzed species.

The software presented in this section called "ViPhy" (www.viphy.org) [BvLH*] was made publicly available and is currently used in the area of biology and bioinformatics by different groups worldwide e.g., in the universities of Edinburgh or California, Berkeley.



(a) (b)

(c) (d)

Figure 4.16.: 16*S* consensus tree and score histograms with background coloring according to the selected elements. a) & b) The selection of a highly conserved pair of organisms. The globally similar *L*18 tree does not contain the marked substructure as one of few. c) & d) The selection of a subtree of seven organisms and their scores across the data set.

5. Extended Comparison Visualization of Large Datasets With Hierarchical Structure

The approach presented in Chapter 4.2 works very well for small trees up to dozens of leaves. If the tree size increases, more and more overlap of tree elements occurs, especially if multiple trees are displayed simultaneously. Optimization of the tree layout only diminishes the problem a little, but if a tree contains a lot more elements than horizontally available pixels, overlapping objects are unpreventable.

Therefore, I propose a new approach called *CloudTrees*, which utilizes the unavoidable overlap to reveal important information (see Section 5.2). Additionally, tree simplification algorithms can be applied as overlay or preprocessing step (or both). As state of the art simplification algorithms do not consider the specific requirements of a comparison scenario, I propose a new approach, specialized on the identification of interesting subtrees in the comparison context (see Section 5.2.2).

5.1. Related Work

This section focuses on the visualization and analysis of *large* trees. Related work for the visualization and the comparison of trees in general can be found in Section 4.1.

Several layout techniques try to position nodes on the screen so that the tree structure is visible even for large trees. To delay negative overplotting effects, space efficient techniques can be applied techniques like treemaps (see Section 4.1.1) or root-centered layouts spreading the tree in all directions. ConeTrees [RMC91] and its extensions [CK95, NCA06] often use a three dimensional representation. This allows a layout without overlapping elements but depending on the rotation and orientation in space, the overlap occurs in its two dimensional screen representation and navigation is more difficult.

There are several two dimensional layout techniques which place the root in the center of the screen and then apply a recursive sub-division of the surrounding space. Circular approaches like in RINGS [TKL02] adding distortion techniques for a better analysis of details or balloon trees [LY07] optimizing the circular layout for unbalanced trees do not use the space very efficiently. Nguyen et al. apply more space efficient techniques using arbitrarily shaped polygons [NH03]. Schulz et al. [SSH09] propose a recursive in partitioning of similar sized areas combining it with point based rendering. They include different interaction techniques like filtering, zooming and a special re-layouting routine for details on demand. This reduces the negative effects of overplotting but for a high space efficiency, the trees need to be balanced and all nodes should have at least 3 children. Balzer and Deussen use different levels of detail to represent large graphs [BD07]. Distortion-based techniques (e.g., fisheye views) allocate more display area to the parts of the tree of higher interest to the user [TAvHS06, CN02, HC04, MGT*03, TS08].

Layout techniques stretching the borders of overplotting and reduce its negative effects but can not prevent it. Simplification-based approaches follow the overview + detail strategy [Shn96] and reduce the amount of details shown in the overview, what makes the analysis much easier [HMM00]. The displayed elements should have a higher relevance to the user than the hidden.

There are several approaches to define the interestingness of elements in general [EF10] and for graphs or trees in special. This can either be defined interactively by the user letting them collapse or expand nodes on demand [SSH09] or selected subtrees are getting more display space [MGT*03]. Algorithmic tree simplification is commonly used for visualization of hierarchical clustering (e.g., [HDM98, CMP09]). The algorithms employ graph and clustering metrics for defining subtrees for simplification. Herman et al. [HDM98] apply the Horton-Strahler Number [Str57] reflecting different aspects of the size of subtrees. The Hierarchical Clustering Explorer [SS02] includes the distances between clusters represented by the elements in the trees for the identification of key nodes. Chen et al. [CMP09] proposed a combination of node distances, leaf count and node balance to extract the overview tree.

All these techniques only work on the structure of a single tree without analyzing similarities and differences to other trees. Therefore, their direct applicability for tree comparison is limited. In particular, interesting subtrees that are found in several trees may be hidden as their structure may not be highly relevant in one single tree.

5.2. Approach: Cloud Trees

In the following, I present a new technique for visualization of large trees specifically focused on tree comparison called “CloudTrees”. First the basis technique for drawing large trees with similarity score information is introduced which then is extended with the visualization of the information on simplification. Finally, CloudTree specific interaction techniques are presented.

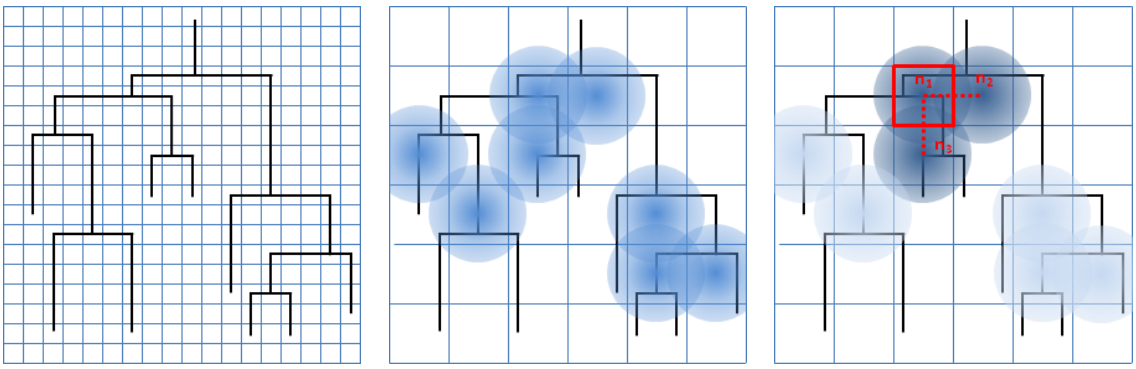
The CloudTree visualization technique aims at providing users with information on similarity scores throughout the tree. Often users search for dissimilar subtrees in generally similar trees. However, this information is often difficult to recognize when visualizing large trees either due to small node size or overplotting, especially if there are more nodes in one line than available pixels (see Figure 5.2a). CloudTree visualization merges the information of the comparison score of multiple nodes in one pixel, i.e., each pixel represents the comparison scores of all superposed elements. This idea is distinctly related to density-based visualization (e.g., CloudLines for time series [KBK11, DLH11]), however my superimposition scheme directly focuses on similarity analysis tasks.

As users may focus on different tasks, a flexible coloring scheme is applied. The users can interactively vary their analysis focus, e.g. putting an emphasis on small, highly similar subtrees (see Figure 5.2b and c).

5.2.1. Cloud Tree Technique

The CloudTree visualization algorithm consists of three steps (see Figure 5.1).

1. **Calculation of the general layout**
   In a first step, the initial layout of the tree is calculated on a sufficiently large grid to avoid overplotting (see Figure 5.1(a)). The layout choice is not relevant for the cloud technique itself. A traditional, orthogonal top-down layout is used, as it is very beneficial for the user’s perception [BKH*11].
2. **Determination of the compact layout**
   For the visualization of the tree, the initial layout is scaled to the available display size. As several graphs need to be shown simultaneously, this step is needed for most of the trees. For this scaling, the center of each node n and edge e is mapped to a corresponding pixel in the lower resolution display space.
3. **Score visualization**
   The similarity score is mapped to color of the nodes which are represented as circles with user defined radius $r(n)$. As color is used for score visualization, node colors of overlapping elements are combined



(a) Layout on arbitrary large grid (b) Projection to actual image size (c) Computation of the pixel values

Figure 5.1.: Schematic steps of the CloudTree calculation. Each blue square represents a pixel. a) Layout of the tree on an arbitrary large grid. b) Projection to the low resolution pixel grid of the visualization and determination of the impact nodes for each pixel. c) Calculation of the pixel value based on the score, size and distance of the impacting nodes.

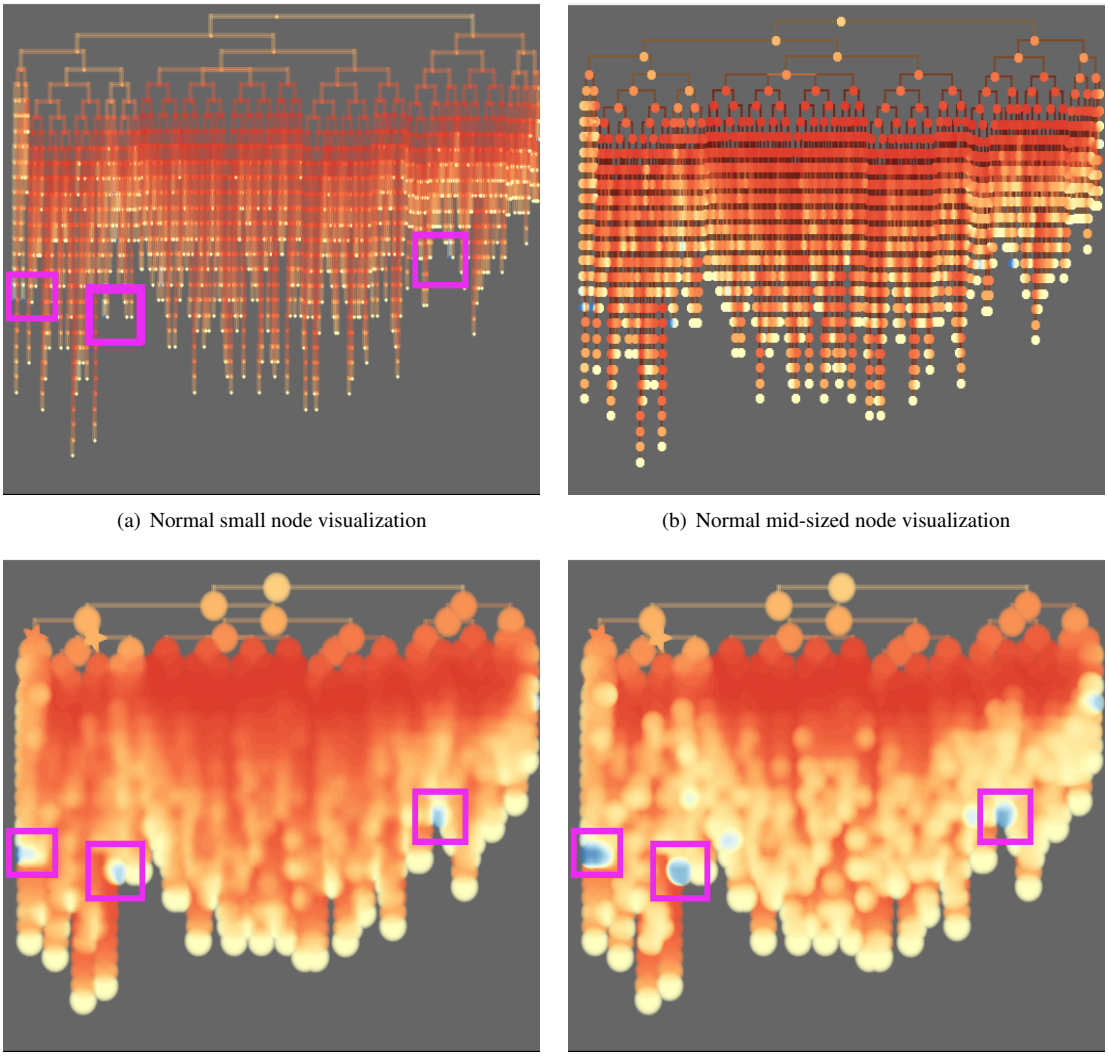
in a special scheme, where I differentiate between inner and leaf nodes. The final color of screen pixels representing inner nodes depends on the underlying visual tree data and the employed color scheme. Each inner node contributes to the color value of all pixels in its so called "impact radius" $r(n)$. For each screen pixel, the final pixel value for coloring is calculated as the sum of the weighted scores of all inner nodes $IN(P_i)$ contributing to the pixel P_i (see Eq. 5.1 and Figure 5.1(c)). The pixels in the resulting image are colored according to their calculated values applying a user defined colormap. In this examples, the same colormap as in section 4.2.4 is used.
Leaf nodes are special as they are identical in all trees and thus always have a similarity score of 1.0. Therefore, they are not included in the above-mentioned calculation and drawn in a user selected color.

The pixel value is calculated as follows:

$$Score(P_i) = \frac{1}{\sum_{n \in N(P_i)} Weight(n)} \cdot \sum_{n \in IN(P_i)} Weight(n) \cdot \widehat{s_e}(n) \tag{5.1}$$

The $weight(\cdot)$ is a function of the node radius $r(n)$, node score $\widehat{s_e}(n)$ and the $d(P_i,n)$ – the distance (e.g., Euclidean) of the center of the contributing node to the respective pixel P_i. For example, Equation 5.2 shows the function for emphasizing small, highly similar trees which otherwise would be hidden (see Figure 5.2(c) and Figure 5.2(d)). The function can be interactively changed. For example, the use of $1-\widehat{s_e}(n)$ instead of $\widehat{s_e}(n)$ in the weight function, would propagate dissimilar regions.

$$Weight(n) = (1 - \frac{d(P_i,n)}{r(n)}) \cdot \widehat{s_e}(n) \cdot (1 - \frac{SubtreeSize(n)}{TreeSize}) \tag{5.2}$$



(a) Normal small node visualization

(b) Normal mid-sized node visualization

(c) CloudTree with linear score contribution

(d) CloudTree with quadratic score contribution

Figure 5.2.: CloudTree visualization of a tree containing 9999 nodes. a) Standard visualization of the tree. For avoiding overplotting, the node diameter was set to 1 pixel. Due to their small size, similarity scores are not readable. b) The node radius was increased to 5 pixel. c) The node radius was increased to 30 pixel. All nodes contribute equally to the pixel score. Some previously invisible, similar regions can be spotted easily (highlighted in purple rectangles). d) Small, highly similar regions are emphasized by scaling the similarity score to its square (see Eq. 5.2).

5.2.2. Simplification for Comparison of Multiple Trees

As the compared trees may be quite large for direct inspection of their similarity, I introduce a new simplification algorithm, which reduces the tree size while considering the similarity among multiple trees (see Figure 5.3 for an illustration).

The idea of tree simplification is to show only the most important structures in the tree and to hide the unimportant parts. The main challenge of simplification approaches is the definition of importance, which is closely related to the analytical tasks (see Section 4). In this case, I concentrate on the comparison of several trees. So the interestingness of a subtree is based on three factors:

1. **Pairwise similarity of subtrees:** Depending on the focus of the analysis, the user is interested in the identification of very similar or dissimilar subtrees. In general, similar substructures in overall dissimilar trees and vice versa are most interesting. In the following, I assume the retrieval of similar substructures.
2. **Size of the subtrees:** Usually, large subtrees with the desired similarity are more interesting than small trees for two reasons. The statistical relevance of large patterns is higher and the collapsing of large subtrees contributes more to the simplification of the tree.
3. **Conservation of the subtrees:** Highly similar substructures which can be found in several trees are more interesting for the user. For example, the information about evolutionary relationships in frequently occurring subtrees is more reliable for biologists.

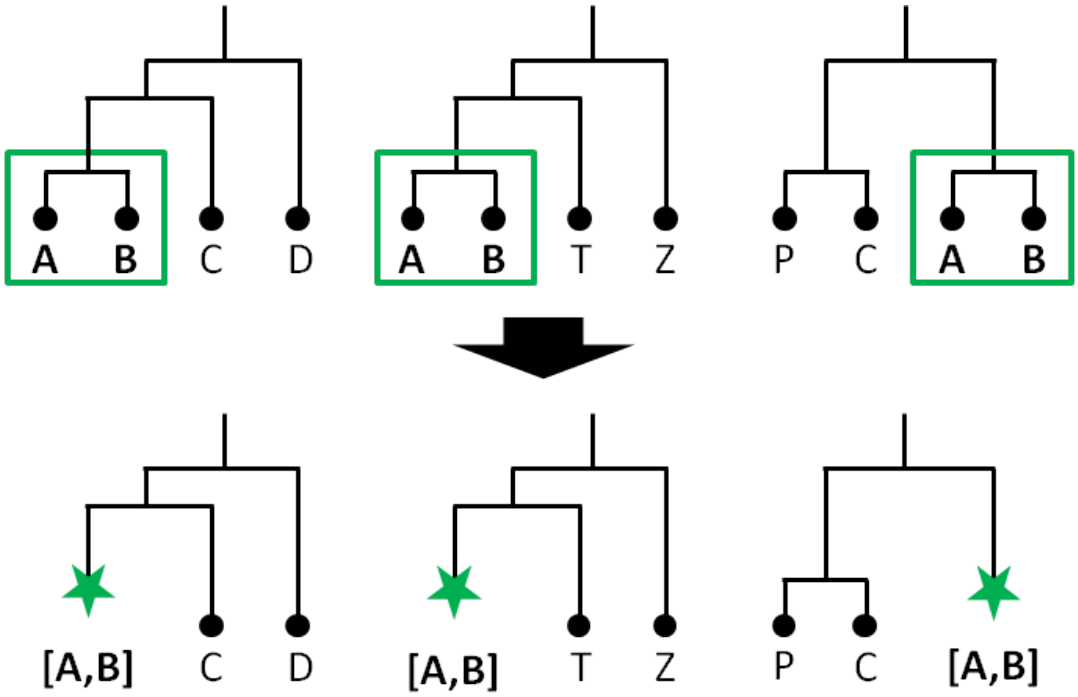


Figure 5.3.: Tree simplification idea. Similar subtrees occurring in many trees are simplified (collapsed) so they can be represented by only one symbol, e.g., a simple mark or glyph.

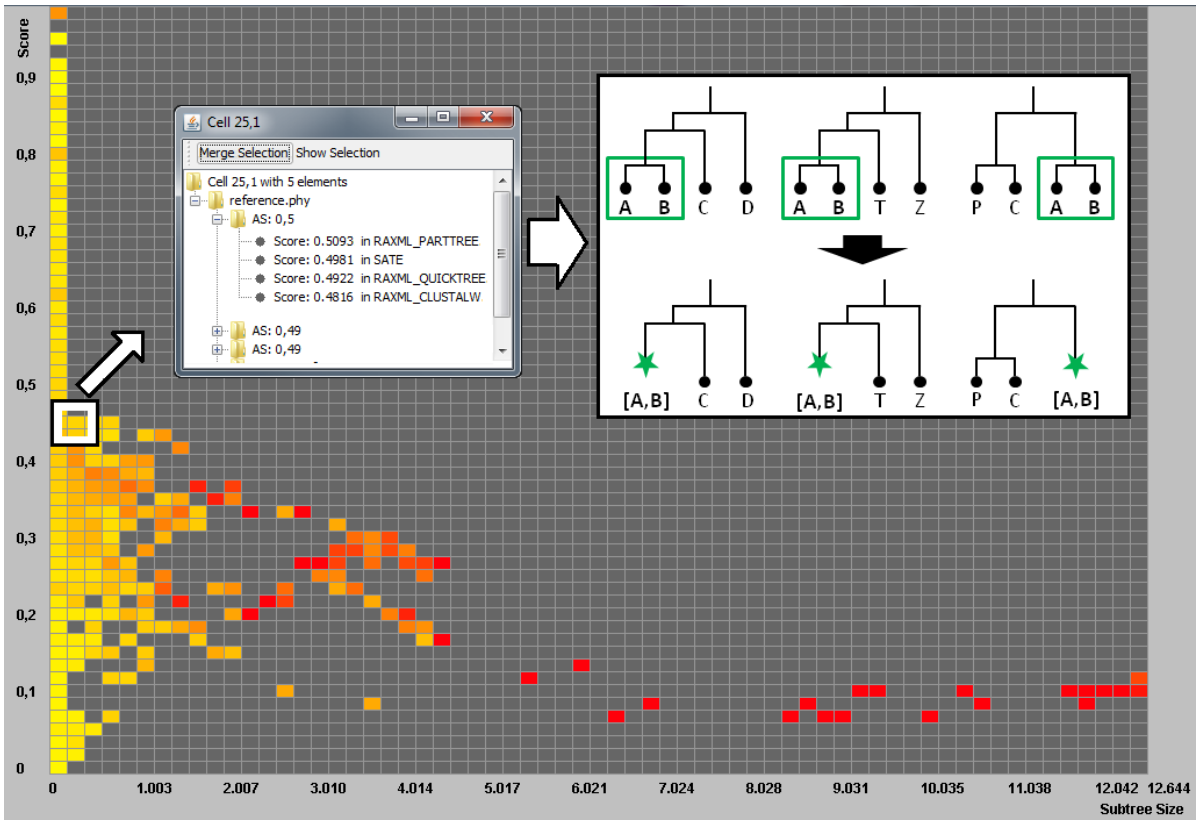


Figure 5.4.: Heatmap visualization of the compared trees including details of a selected cell. The contained subtrees can be merged on demand.

The conclusion is that the most relevant subtree for simplification (in the following called "Subtree Cluster" (SC)) has a very high score, is large and can be found in many trees throughout the dataset. However, in real-world applications, there is a trade-off among these three factors. For example, a higher score for the same tree size is better but, at the same time, the equal score for a larger tree is more relevant. So it is difficult to automatically determine which structures should be collapsed. Therefore, I rely on user expertise in weighting these factors.

For simplification support, we propose a heatmap-based visualization showing the three above-mentioned criteria. It enables the user to define her interest (see Figure 5.4). In the view, each subtree of the data set is treated as an object assigned to a cell of the heatmap according to its score (x-axis) and size (y-axis). The cell color reflects the number of contained subtrees. There are two possibilities for color normalization – global vs. local along tree sizes. Both normalization and heatmap resolution can be chosen interactively. Nevertheless, the local column-wise normalization is preferable as it allows a better comparison among equally sized trees.

Detailed information about subtrees in a selected heatmap cell can be displayed on demand (see Figure 5.4). It includes data on their pairwise score distribution with all subtrees in the dataset ordered by size and score. In this way, the user can iteratively identify interesting structures and form Subtree Clusters (SC) for simplification. If the user selects a set of trees as SC, all contained subtrees with a lower consensus score are removed from the heatmap as they are now represented by the SC. Higher scored subtrees remain as they might contain additional valuable information for the analyst. Usually, the user starts from the upper right corner of the heatmap (trees with both high score and large size) and continues to the left. Additionally, an arbitrary set of user selected subtrees from a cell can for be displayed for detailed visual exploration.

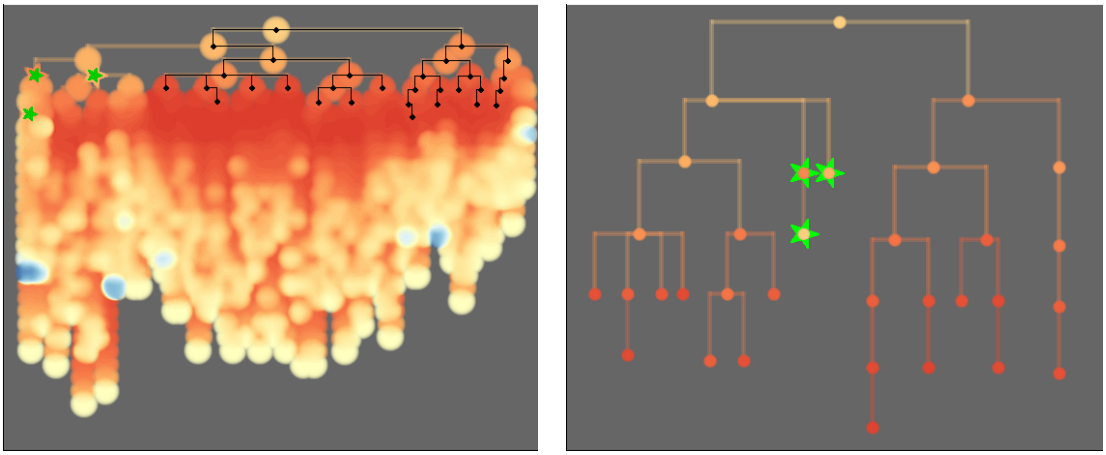
Additional simplification on demand The resulting trees from the above-described simplification may still be relatively large. Therefore, the user has an option to further simplify the trees using the techniques for individual trees referenced in Section 5.1. In this use-cases, the Strahler Number [Str57] is employed as simplification measure, as it is independent of the tree structure and requires no node distances. The threshold is user-defined and nodes representing subtree clusters are always preserved. If a father of a SC is discarded during the simplification process, the former grandfather becomes the new father.

5.2.3. Extended Cloud Visualization for Simplified Trees

Simplified trees can be displayed as overlay above the CloudTree. By this, we ensure the visibility of the subtrees which were identified as interesting during the simplification. Every node of the simplified tree is positioned on the respective pixel on the CloudTree. Because the original trees can be simplified in two ways, each simplification is represented differently in the view, so that the user can distinguish the type of collapsing applied (see Figure 5.5(a)).

1. Comparison-based simplification: Interesting substructures which were interactively defined using the heat map view are represented by a mark (a green star in my case).
2. Structure-based simplification: All other nodes which were not pruned by the Strahler simplification process are represented as black nodes in the overlay.

As an alternative view, to increase space efficiency, the simplified tree without the CloudTree can be shown. It then uses the whole available display space (see Figure 5.5(b)).

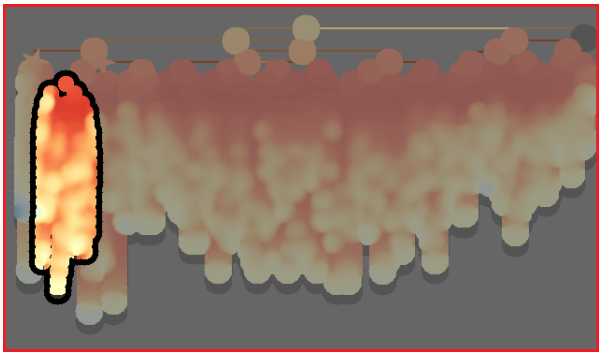


(a) (b)

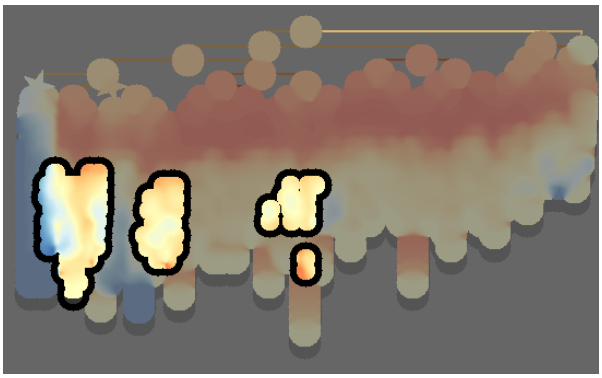
Figure 5.5.: a) CloudTree with an additional overlay of the in the heatmap view defined interesting subtrees (green stars) and the core structure after Strahler simplification. b) Presents only the simplified tree as a compact visualization.

5.2.4. Interaction in CloudTrees

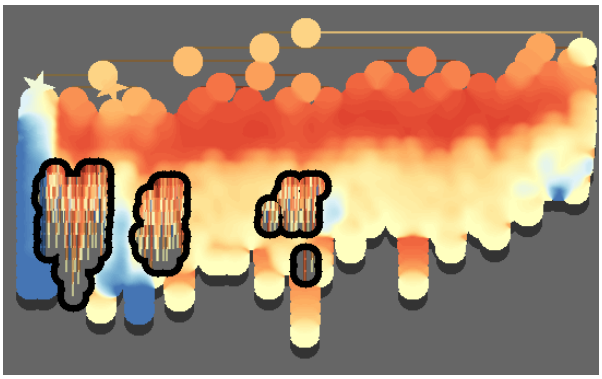
Interaction with the trees mainly aims on the selection of regions of interest for several analytical goals. For comparative tasks, the interest focuses on the distribution of the highlighted elements of one selected tree compared to all other trees. To give an example, let us assume the existence of two similar but not identical subtrees in two compared trees. The question which usually arises is: where are the elements of one subtree which discriminates it from the other subtree in the compared tree. For the selection of subtrees, it needs to determined which exact node is selected by the user. If more than one element is represented by a single pixel, interaction is complicated as unique manual selection will not be possible. In the CloudTree visualization, the users can choose one of several interaction modes that fits best to their analytical task. The interaction modes include the selection of all contained elements, selection of only the best or the worst element or selection of the element with the largest or smallest subtree. The selected node and its subtree are highlighted in the particular tree and, if present, also in all other trees in the visualization (please see the video “CloudTrees” on the appended DVD). The selected areas can either be highlighted, keeping the CloudTree coloring or rendered transparent to reveal more of the underlying tree structure (see Figure 5.6). For a detailed analysis of a single subtree, the layout can be adjusted on demand. More space is assigned to the selected subtree on expense of increasing the density of the unselected tree. Alternatively, subtrees can be displayed in separate windows.



(a) Selection in reference tree



(b) Highlighted distribution in compared tree



(c) Transparent distribution in compared tree

Figure 5.6.: Highlighting in the CloudTree: a) A subtree in the reference tree is selected and highlighted. b) All in (a) selected elements are highlighted in the compared tree. In this case, they are distributed to 3 areas. c) An alternative highlight mode reveals the underlying tree structure.

5.2.5. Visual Interface

The visual-interactive design offers four linked views representing various aspects of the analysis (see Figure 5.7).

View 1: Shows the similarity matrix reflecting global pairwise similarity between all trees and offers the selection of one reference tree for detailed comparison with all other trees.

View 2: Provides information on score distribution of all nodes in each tree when compared with the selected reference tree. The distribution pattern allows to see whether there are parts of a tree with better match then other parts.

View 3: Shows the so called "consensus tree" view of the reference tree selected in view 1. It shows consensus similarity scores (see Section 4.2.3.2) thereby offering a compact comparison of the reference tree with all other trees.

View 4: Shows a user-selectable number of trees chosen for a detailed analysis with the nodes color-coded by their similarity scores.

The linked views of the tree comparisons allow for the highlighting of selected structures and their best matches in all visible trees (see Section 5.2.4).

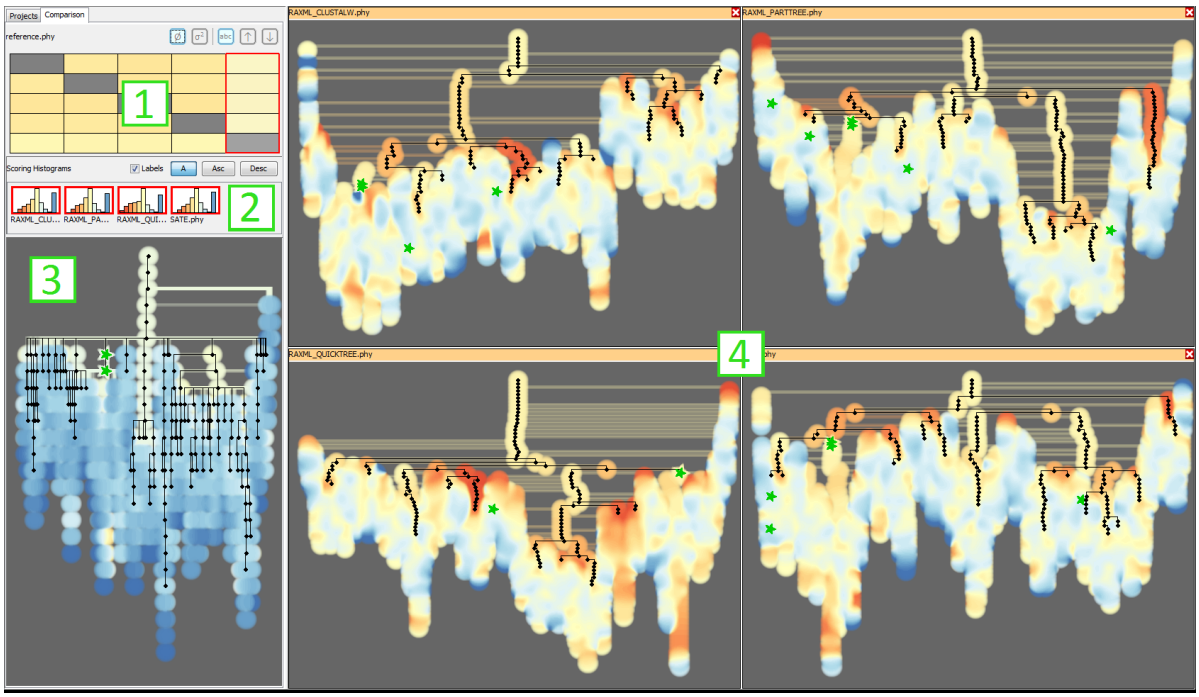


Figure 5.7.: Tree comparison view. (1) The similarity matrix reflecting the pairwise overall similarity between the trees. (2) Score distribution in trees compared to the reference. (3) Reference tree. Node color represents the average score of the nodes throughout the data set. (4) Selected trees, node color indicates subtree similarity to the reference tree shown in (3). Star shaped nodes and black overlay nodes represent the respective simplified trees.

5.3. Application to Chemical and Phylogenetic Data

In the following, the usage of my system for visual comparison of multiple large trees is demonstrated in two different use cases:
1) Comparison of descriptors for chemical components based on hierarchical clustering
2) Analysis of the relationship amongst species conducted through a comparison of phylogenetic trees.

5.3.1. Chemical Descriptor Comparison

The analysis of vast amounts of chemical data is a common task in chemistry, biology and pharmacy. For example, high throughput screenings (HTS) automatically conducts thousands of experiments, e.g. testing the molecule activity against a target compound. All molecules above a certain activity threshold are selected and the analyst's task is to categorize these molecules and reveal the underlying systematics. This is done using molecule descriptors. Many descriptors can be calculated for each molecule, some are one dimensional for instance describing the weight or the charge of a molecule, other descriptors are multidimensional like the chi-chain (10-D) or the fingerprint-based descriptor (with up to several hundred dimensions) [LG07]. One goal of this exploratory analysis is to find connections between groups of molecules whose descriptors exhibit similar characteristics. In this example, the used dataset consists of 20 descriptors for 5000 molecules. For hierarchical clustering, average linkage with two distance measures is applied: Euclidean distance for the floating value descriptors and the Tanimoto index [RT60] for the binary fingerprint descriptors. As a result, each descriptor is represented by a tree with 9999 elements.

After the calculation of the similarity scores for the trees, the results are visualized using the heatmap view (see Figure 5.9). It shows how the normalization facilitates correct interpretation of the data. Without normalization, some quite large trees with very high scores can be observed (see Figure 5.9(a)). Normalization, however, reveals that the most of the scores are around the theoretical minimum and that the trees are not as interesting as initially assumed (see Figure 5.9(b)). The outlying large trees with high similarity are formed by a group of three trees based on the descriptors for the sum of atomic polarizabilities, weight and normalized polarizabilities. The polarizability heavily depends on the size of the molecule, so the number of atoms (which is roughly correlated to the molecule weight) is used for the normalization. As the similarity of the mentioned three trees on a global level is not surprising, a number of small trees (40–120 elements) with high similarity value (≥ 0.6) is selected. (see Figure 5.8). In this selection, two groups of descriptors often appear together: The first consists of polarization, normalized polarization and weight descriptors. The second group comprises descriptors for size, wienernumber, kappashape and eccentric connectivity.

After the identification of interesting substructures we take a closer look using the cloud-tree visualization. The dendrogram resulting from clustering in the polarization descriptor space is selected as reference tree, because it already stood out in the simplification phase. As expected, the normalized *polarization* and *weight* trees are most similar (top two trees in Figure 5.10(a)). The two trees on the bottom represent the second group of previously noticed descriptors.

An additional overview of the simplified trees (small black nodes & green stars) reveals some interesting findings. A highly similar region, which was not discovered before and is not captured by the simplified trees (see Figure 5.10(a) purple rectangle), can now easily be identified. On the other hand, there are no noticeable patterns visible around the stars in the top hierarchies (see Figure 5.10(a) pink rectangle). Without the previous simplification step, these interesting subtree clusters would be missed. None of the stars is part of the simplified trees, as they are not connected to the black Strahler overlay nodes. If the analysts would rely only on the simplified trees, they would not find the previously identified similar regions represented by the stars (see Figure 5.10(a)). How-

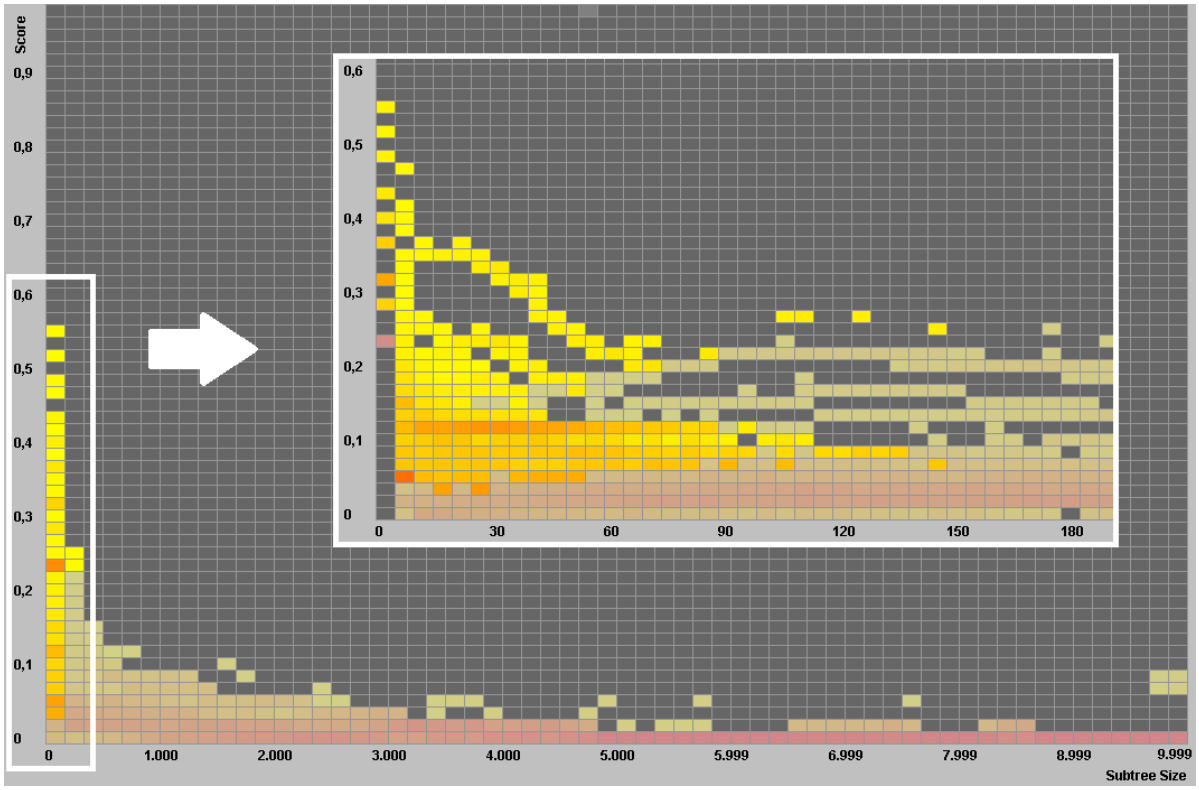
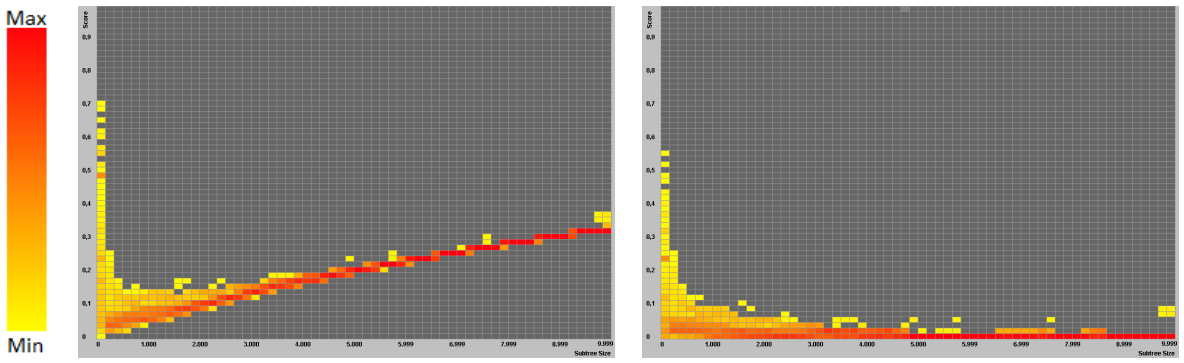


Figure 5.8.: Heatmap view, where each cell represents the average score of the contained subtrees. Cells containing a subtree with an maximum score > 0.6 are highlighted. The overlay shows a zoomed in view of the subtrees with < 190 elements.



(a) Score distribution without normalization

(b) Score distribution with normalization

Figure 5.9.: Heatmap visualization of the comparison score (x-axis) versus subtree size (y-axis) of the chemical compound dataset. The color of each cell reflects the number of subtrees in one cell. a) Scores without normalization. b) Scores with normalization based on theoretic score minimum values. A larger version of these images can be found in Figure 4.7.

ever, the reduced tree offers enough orientation for a further analysis of the subtree-clusters (see Figure 5.10(b)). Some of the trees are very unbalanced and show finger-like patterns (see Figure 5.10(c), bottom row). This pattern occurs, if many molecules are described with the same value e.g. using the ring-count descriptor. Many molecules have the same number of aromatic rings.

5.3.2. Phylogenetic Tree Analysis

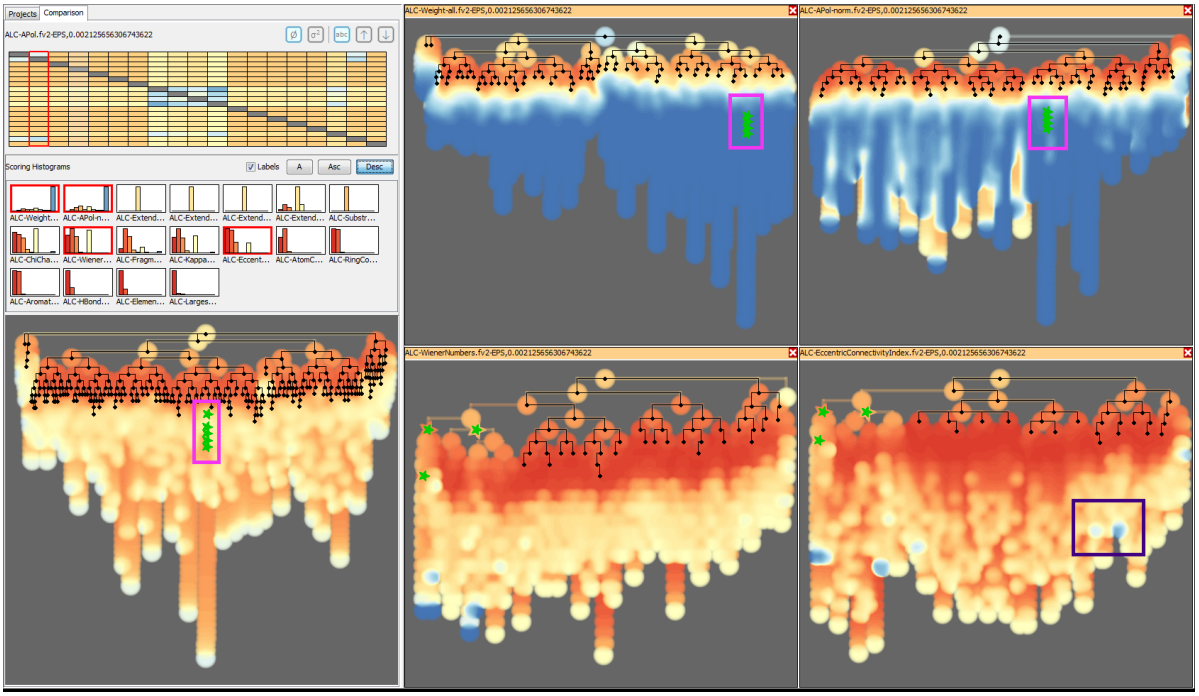
This use case focuses on comparison of phylogenetic trees. The trees describe evolutionary relationships among species and are commonly used in evolutionary research. For example, the identification of close relationships of bacteria can help to generalize knowledge about a specific species, which can support the extension of medical treatment to other bacteria in the group. For a detailed description of the calculation of phylogenetic trees and the tasks in the phylogenetic tree comparison see Section 4.3.1. In this example, a dataset from Liu et al. [LLW10] is analyzed describing 5 trees, each containing 6323 species. Each tree was calculated using a different algorithm (for details please see [LLW10]). One of the trees was hand-curated, therefore it serves as the reference tree for this analysis example. Please note, that this is an n-ary tree which has significantly less inner nodes.

The heatmap view (see Figure 5.11) shows that the similarity of the subtrees is substantially higher scored than in the first use case. This is not fully unexpected, because the subtrees generated by the different algorithms illustrate the evolutionary relationship of the same species.

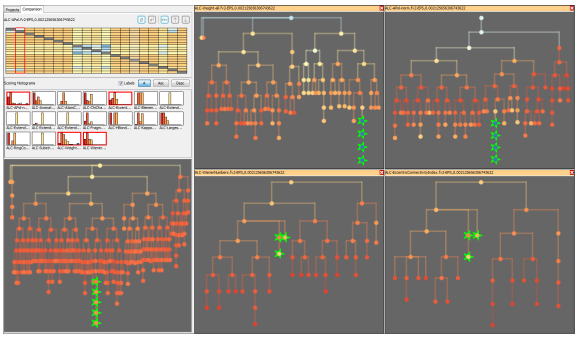
For a more detailed look at the data, Figure 5.11 shows a zoomed view on the trees up to 60 nodes. The highlighted cells contain highly similar subtrees (with a score > 0.8). These are mostly subtrees with only three elements – the smallest possible tree.

Different interesting subtree clusters are selected, some rather small with high similarity scores and two larger ones, each having around 1200 elements with a score of ca. 0.48. The resulting trees still contain 8000 to 10500 nodes.

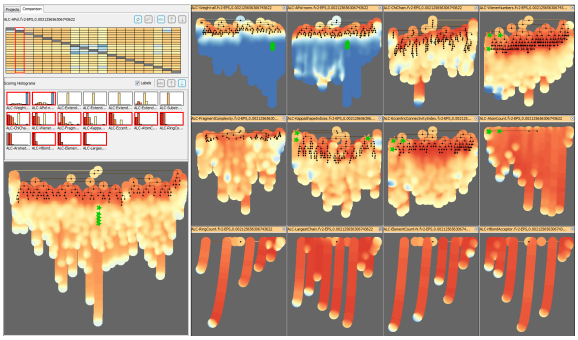
The following CloudTree visualization of all simplified trees shows a very heterogeneous score distribution of the compared trees, while the reference consensus tree presents a good overall similarity (see Figure 5.7).



(a) Cloudtrees including overlay



(b) Compact visualization of simplified trees



(c) Comparison with many trees

Figure 5.10.: Comparison visualization of four trees to a reference tree. a) Cloud-tree visualization with an additional overlay of the reduced trees. The pink rectangle highlights subtree clusters (stars), the purple rectangle highlights a pattern which is only visible in the cloud trees. b) The simplified trees offer enough orientation for a further analysis of the subtree-clusters. c) Figure showing 12+1 trees for detailed analysis. Trees are sorted from similar to dissimilar to the reference tree.

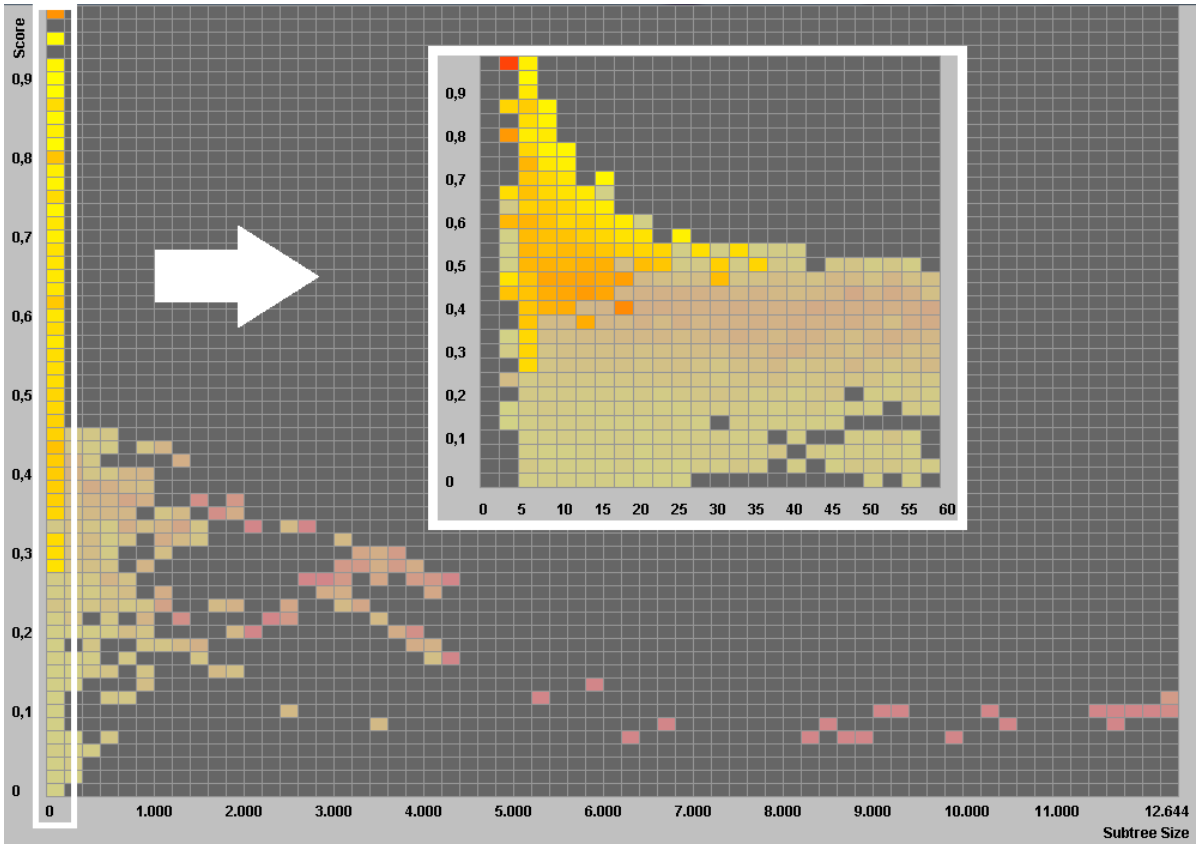


Figure 5.11.: Heatmap visualization of the phylogenetic trees. Cells containing trees with a maximum pairwise similarity > 0.8 are highlighted. The overlay shows a zoomed in view of the subtrees with < 60 elements.

This clearly shows that none of the algorithms really calculates the optimal result and the deviations may occur at various positions and to various extents throughout the compared trees. On the selection of a subtree in the reference tree, all contained elements are highlighted in all trees. Due to the high overall similarity, it is not surprising that these elements are usually located in a small area of the respective tree, only one tree exhibits some outliers. This means some of the species which are stated to have a close evolutionary relationship are far distanced in this tree. This is an interesting finding for the biologist and starting point for further investigations.

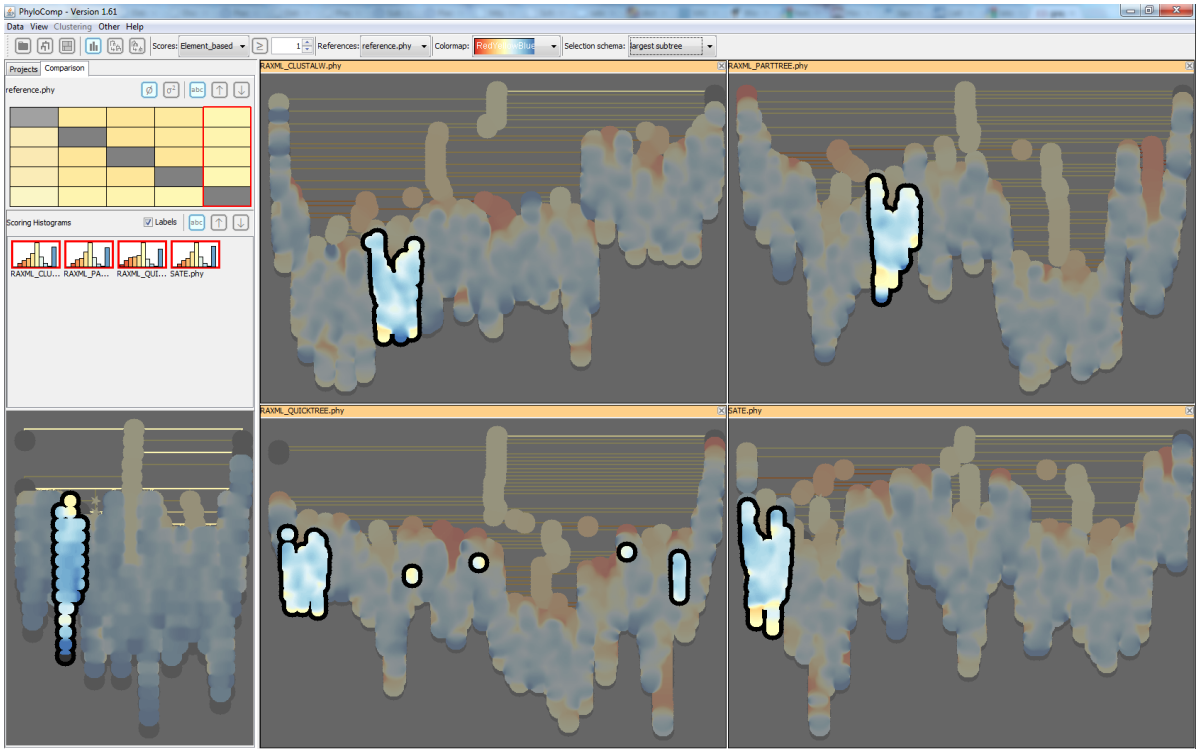


Figure 5.12.: This figure shows the interaction with CloudTrees. A subtree in the reference tree (bottom left) is selected. The contained elements are highlighted in all shown trees. One of the compared trees shows an divergent pattern of the otherwise compact distribution.

6. Visual Comparison of Datasets with Unknown Structure

As shown in the previous chapter, the comparison of different descriptors leads to more insight during the exploratory analysis. The technique proposed in chapter 4 is based on hierarchically organized datasets or at least on a hierarchic representation of the dataset (e.g., using hierarchical clustering). In contrast to this, in the following chapter, I will present an approach which does not rely on a specific structure of the data.

To involve the users and their knowledge in the explorative analysis process, a visual representation of the descriptors is needed. To allow a consistent visualization, this representation has to be independent of the dimensionality of the descriptor. Therefore, I use different common dimensionality reduction methods like Principal Component Analysis (PCA), Multidimensional Scaling (MDS) or self-organizing Maps (SOMs), projecting the data of each descriptor into two dimensions. I present a novel color-based visualization technique to compare these 2D projections. The technique allows to identify global similarity as well as local patterns and works for different representations (e.g., scatterplots or grid based techniques like SOMs) (see Section 6.2.2). Additionally, a representation of the projection/mapping quality can be included in the visualization (see Section 6.2.4.1). To guide the user through the analytical process, I propose an automatic calculation of the descriptors similarity adapting various states applying techniques normally used to measure projection quality (see Section 6.2.1).

The work presented in this chapter is partially based on [BHvLF12, SvLB10] and [BvLBS11].

6.1. Related Work

This work is related to approaches for the dimensionality reduction of high dimensional datasets e.g., for visualization or model construction.. The data should be represented as accurate and at the same time as simple as possible. These approaches can be divided in two classes of algorithms: feature extraction and feature selection. Feature extraction methods project the data in a low dimensional, extracted space, by minimizing or maximizing algorithm specific criteria. The new, extracted descriptors do not necessarily have a direct connection to the input descriptors. Feature selection approaches construct the low dimensional space as a combination of few selected input descriptors. For this selection, many techniques for the evaluation of single descriptors or features have been proposed. Each descriptor is evaluated individually using different metrics e.g. different sorts of entropy (Shannon entropy [SW49], entropy of distances between data points [DCSL02]), gauss distribution similarity [DF84] or mutual information. For the comparison of two features or a feature input and a classification, different correlation coefficients can be calculated [Hal99], e.g. Pearson's or Spearman's correlation coefficient [OHS05]. For a broader overview, please see the survey of Guyon and Elisseeff [GE03].

All of these scores calculate a global value for each descriptor, but no local, object-wise scores. Thus, it is not distinguishable if e.g. a medium score is caused by a general medium divergence or by a high variation of few objects. In contrast, object distance and topology based methods calculate a score for each point, whereas the score for the whole descriptors is a combination of them. These approaches are used for the feature selection based on dimensionality reduction e.g. preserving the local distances [HCN06] or keeping the nearest

neighbor distances and simultaneously maximizing the global pairwise distances [WS06]. Alternatively, these neighborhood approaches are used for the evaluation of dimensionality reduction. More details can be found in Section 6.2.1 where I compare different scores for their capabilities for descriptor comparison.

6.2. Approach

My approach combines two aspects. First, pairwise distances between all descriptors are calculated automatically, which supports the handling of large datasets. Based on these distances, groups of similar descriptors are identified (e.g. by clustering of the descriptors) and potentially interesting descriptors can be recommended to the user for further analysis. The second component is the interactive comparison visualization. The visualization allows the analyst not only to review the results of the automatic comparison, furthermore it allows a very detailed inspection. It helps for example to identify local patterns. To give examples and evaluate the effects of comparison algorithms and visualizations, I introduce a synthetic test dataset providing a solid ground truth. For its fast creation, I introduce PCDC (Parallel Coordinates Data Creation), which offers a parallel coordinates based interface to allow a detailed control of the sampling parameters [BHvLF12]. The datasets are created based on 4 different models containing zero to three gausian distributed clusters each. Objects which are not within a defined cluster are distributed equally in the free (i.e. not occupied by a cluster) value range. The objects were created multiple times for each sampling model. After an initialization, 25%, 50%, 75% and 100% of the objects where re-sampled resulting in 5 datasets per model. Each of the 20 datasets contains the same 4500 objects and thereby represents a different descriptor. For details, see Table 6.2 and Figure 6.1

The idea behind the creation schema is twofold. On the one hand to compare the influence of re-sampling within each model and on the other hand the comparison of different models. More variables will be covered in future work (see Section 11.1).

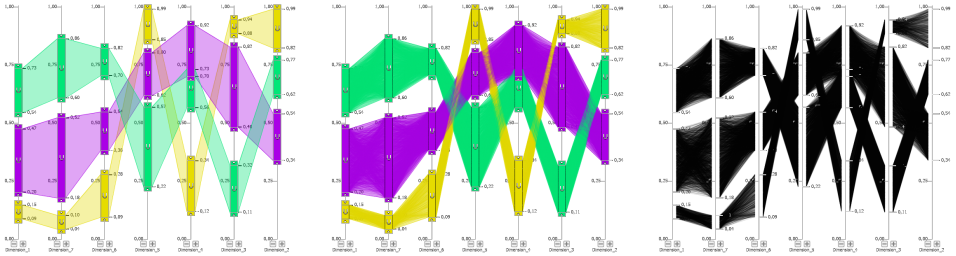
6.2.1. Comparison Measures

In this approach, I rely on a pairwise comparison of the descriptors. Although it would be possible to build a descriptor for a descriptor e.g. measuring properties like entropy or density, its usefulness in a comparison scenario would be limited. These measures are too general and unspecific to provide a sustained comparison base because different object distributions could have similar values. The measures I use are commonly employed to assess the projection quality of different algorithms. They compare the structure of a high and a low (typically two or three) dimensional dataset represented by some sort of neighborhood relationships of the contained instances [Koh82]. This neighborhood is usually calculated measuring the pairwise distances between all objects in the dataset using an appropriate distance function. The comparison of two datasets containing the same objects can now be calculated based on the raw distance or on a local neighborhood ranking. Both approaches have their advantages (see Figure 6.2), but the most recent approaches rely on topology based measures as they have a broader applicability.

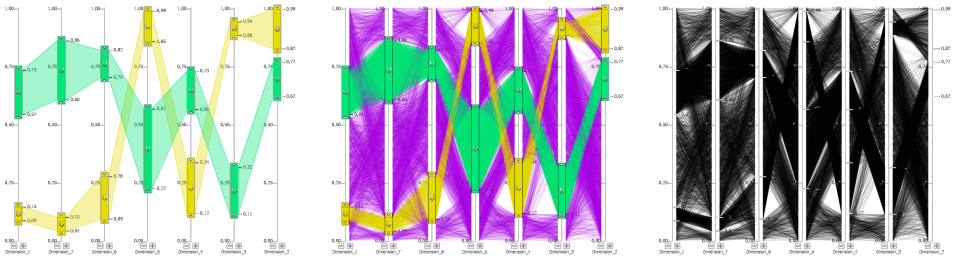
ID	Dimensions	Classes[a]	Cluster	Noise	Resampling	Name	Model
1	10	3	C1,C2,C3		Initial	M1S1	
2	10	3	C1,C2,C3		25%	M1S2	
3	10	3	C1,C2,C3		50%	M1S3	M1
4	10	3	C1,C2,C3		75%	M1S4	
5	10	3	C1,C2,C3		100%	M1S5	
6	10	3	C1,C3	C2	Initial	M2S1	
7	10	3	C1,C3	C2	25%	M2S2	
8	10	3	C1,C3	C2	50%	M2S3	M2
9	10	3	C1,C3	C2	75%	M2S4	
10	10	3	C1,C3	C2	100%	M2S5	
11	10	3	C3	C1,C2	Initial	M3S1	
12	10	3	C3	C1,C2	25%	M3S2	
13	10	3	C3	C1,C2	50%	M3S3	M3
14	10	3	C3	C1,C2	75%	M3S4	
15	10	3	C3	C1,C2	100%	M3S5	
16	10	3		C1,C2,C3	Initial	M4S1	
17	10	3		C1,C2,C3	25%	M4S2	
18	10	3		C1,C2,C3	50%	M4S3	M4
19	10	3		C1,C2,C3	75%	M4S4	
20	10	3		C1,C2,C3	100%	M4S5	

[a] C1,C2:2000 objects; C3: 500 objects

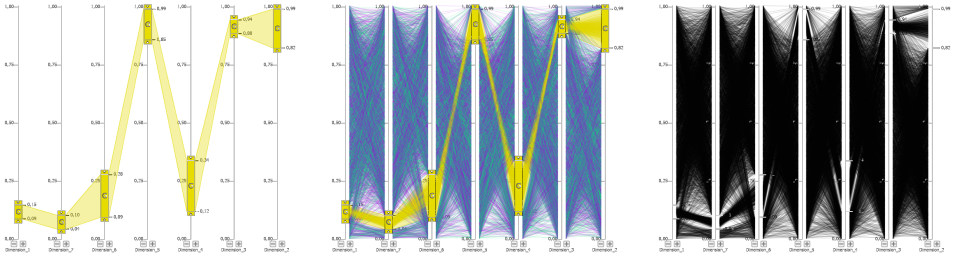
Table 6.1.: 20 synthetic datasets were created according to four different sampling models (M1, M2, M3, M4). Each model has ten dimensions and contains 4500 objects assigned to the three classes C1, C2 and C3. The objects of each class form either a cluster with gaussian distribution or are equally distributed as noise.



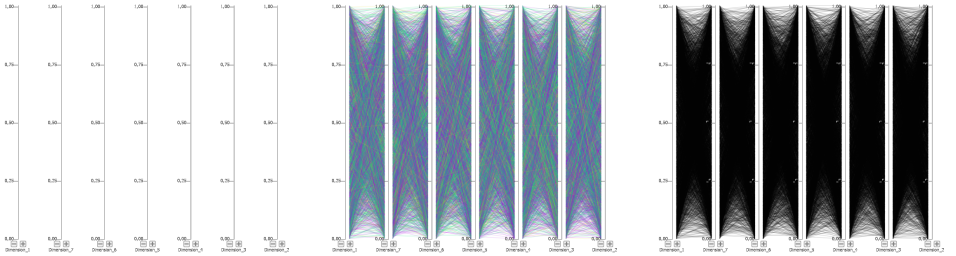
(a) sampling model M1



(b) sampling model M2



(c) sampling model M3



(d) sampling model M4

Figure 6.1.: Visualization of the sampling models and the sampled data. Each row represents one model defining the distribution of the objects of the three classes C1 (purple), C2 (green) and C3 (yellow). The first column shows the models defined in PCDC, the second includes colored sampled data and the third column contains the sampled data without labels.

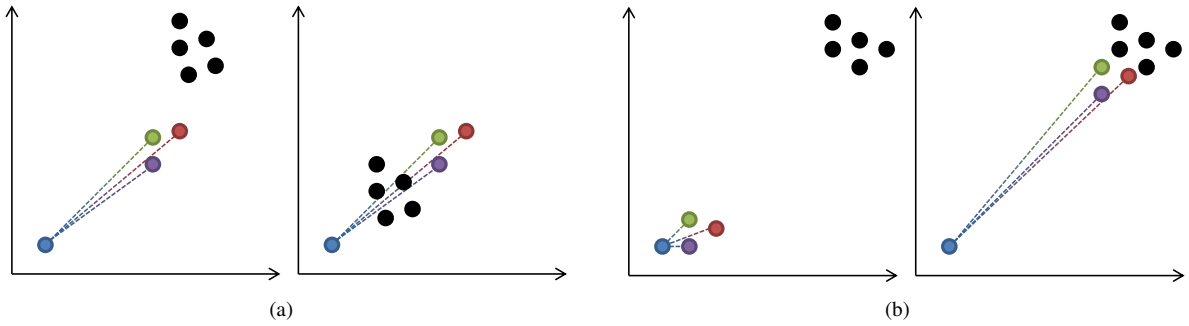


Figure 6.2.: Comparison of topology and distance-based measures. a) In both two dimensional distributions, the distance between the blue and the other colored objects is identical but the topology changes. Topology based measures would take the difference between the point clouds into account, whereas distance based measures would not. b) In this scenario, the topology is constant but the distance changes. Distance based measures are advantageous to topology based ones.

As each comparison algorithm emphasizes slightly other criteria, I implemented seven representative state of the art algorithms covering three families of topology based approaches as well as a distance based approach. Each of them depends of a respective parameter k representing the neighborhood rank which lies between 1 and the number of objects in the dataset.

- Trustworthiness and continuity (T&C, topology based) [VK01] measures changes of the neighborhood from within to outside of k. These large changes in topology indicate big dissimilarities of the compared descriptors. In contrast to the other scores, no changes of the topology within the range of k are regarded. Especially for large k, the result differs from other scores like LCMC, PP or MRRE (see Figure 6.3).
- Local continuity metacriterion (LCMC, topology based) [CB09] measures neighborhood changes within the range of k. For small k, changes within k can be regarded as insignificant or as only slight difference between the compared entities.
- Mean relative rank errors (MRREs, topology based) [LV07] measure both types of changes in topology.
- B_nx [LV09] (topology based) subtracts the number of neighbors departing and approaching within the range of k. Thereby, it is possible to identify a direction in the change of topology.
- Our score, Projection Precision (PP, distance based) [SvLB10] measures the difference in the distance between the k nearest neighbors in one descriptor compared to the distances of the same objects in an other descriptor.

Rank based as well as the distance based approaches heavily depend on the selected parameter k, defining the number of neighbors included in the score calculation. A comparison using small k can result in a good score, even if the descriptors are globally dissimilar whereas local patterns could be missed out using a large k. To enable the analyst to identify the differences between the different algorithms or parameter settings, a hierarchy for each algorithms/parameter combination was calculated using hierarchical clustering (average linkage). Each tree contains the same descriptors represented by the leaves, so my approach for comparing trees could be applied (see Section 4).

The result shows, that all measures which sum up topology changes within k (LCMC, MRREs and PP) produce similar results (see Figure 6.3). For small k, T&C are similar, too, but with increasing k the similarity decreases as

the disparities of the algorithms (within k vs. outside k changes) have a stronger impact. Furthermore, the special approach of B_mx calculating the direction of changes is clearly visible as it produces dissimilar results compared to all other measures. In consequence, PP is used as standard comparison measure. Changes in the neighboring topology are often more interesting for the user as this is e.g. a relevant information for cluster assignment. Additionally, as the score is calculated per object, local changes can be visualized (see Section 6.2.4.1).

For the calculation of the computational complexity, n is assumed as the number of objects with k=n and m is the number of compared descriptors. The calculation can be split in two parts. First, the ranking or distance matrices for all descriptors are calculated in $O(m*n^2*log(n))$. Second, for the comparison differences in distances or rather coranking matrices are calculated and evaluated. Both is done in $O(n^2)$ for each of the $O(m^2)$ pairwise descriptor comparisons. This results in a total worst case runtime of $O(m*n^2*log(n)+m^2*n^2)$.

All implementations of topology based measures employ an idea of Lee and Verleysen [LV09]. They formulated a so called coranking matrix which is a joint histogram of two ranking matrices of two compared descriptors. Different areas of this coranking matrix can then be evaluated building the basis for different comparison measures. This enhances the constant proportion of the calculation. The single-threaded calculation of all mentioned measures for a dataset containing 20-30 descriptors for 10000 objects each takes under five minutes.

6.2.2. Comparison Visualization

Each descriptor and thus each visualization of the described objects may have a different topology. Often, this information is mapped to the relative positions of the objects in space. The idea of my approach is to combine two differing topology information in one visualization. One is, as usually, mapped to position, the other is represented by color (see Fig 6.4 for a toy example combining two 1D representations). By this, the topology information of the two descriptors can be merged in one visualization which alleviates their comparison.

The visualization of the descriptor comparison builds upon a two-dimensional representation of the data which can either be displayed as scatterplot or grid-based visualization to reduce overplotting. The projection of the data relies on the topology preservation property of dimension reduction algorithms. From a variety of approaches, a selection containing PCA, Kernel PCA, MDS, Sammons Mapping and SOM is considered.

For descriptor comparison, I extend the scatterplot and grid-based (SOM) visualization with similarity information. I propose a dedicated color visualization scheme for pairwise comparison of two descriptors in one single view. The data of the reference descriptor D_a are used for color coding of the data of the compared descriptor D_b. A two-dimensional colormap is applied for the color coding as the individual views of the data are in 2D (see Section 6.2.3). The objects of the compared descriptors are then shown in the two-dimensional projected space. The color distribution of the objects in the compared space indicates the similarity of the two descriptors (see Figures 6.7 and 6.10). As the color coding is based on colormaps with a homogeneous color gradient, objects which are nearby in the reference are similarly colored. If neighbouring points in the comparison visualization have similar colors too, then these objects are neighbors in both compared descriptors. If two neighboring points in the comparison visualization have dissimilar colors, they are far distant in the reference (see Figure 6.5).

6.2.3. Two-Dimensional Color Maps

Coloring the data in a two-dimensional space, such as in a scatterplot or SOM grid, is a challenging task. It is difficult to balance the following beneficial properties at the same time: a perceptual linearity of the color space, a high color resolution and the preservation of all pairwise prototype distances [KVK99]. For two-dimensional coloring, in particular SOM coloring, a number of color-based visualization techniques were proposed [KK98, Him98, KVK99, KVK00, Him00]. The idea is to apply high-contrast color space to illustrate the SOM grids

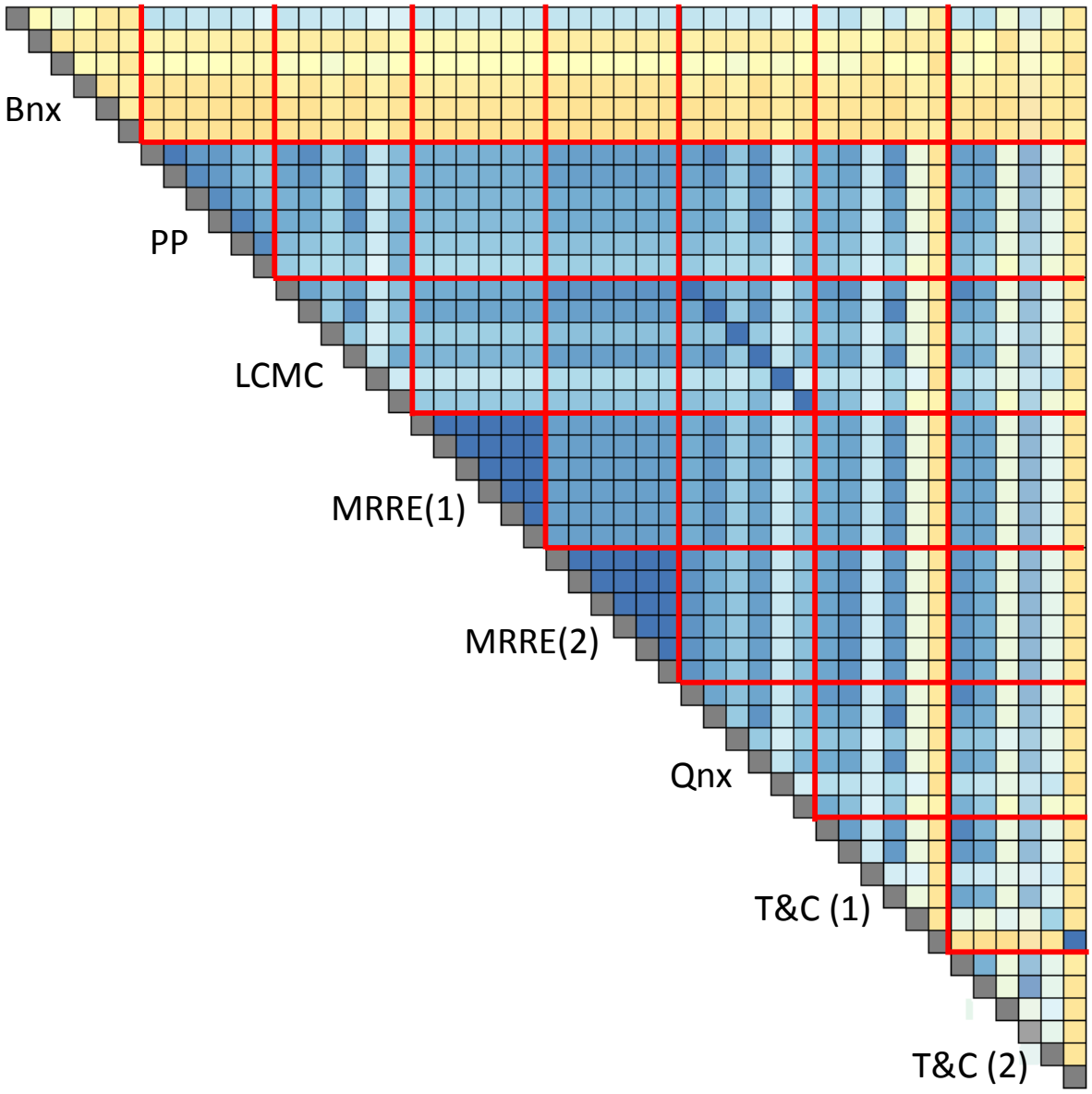


Figure 6.3.: This figure shows a comparison of the different algorithms. Each field represents a comparison of the distance matrix of all pairwise comparisons of the 20 artificial datasets. The color reflects the calculated similarity score which changes using different k and different algorithms to create these matrices. Each algorithm was run with six different k (1%, 10%, 25%, 50%, 75% and 100% of the data size)

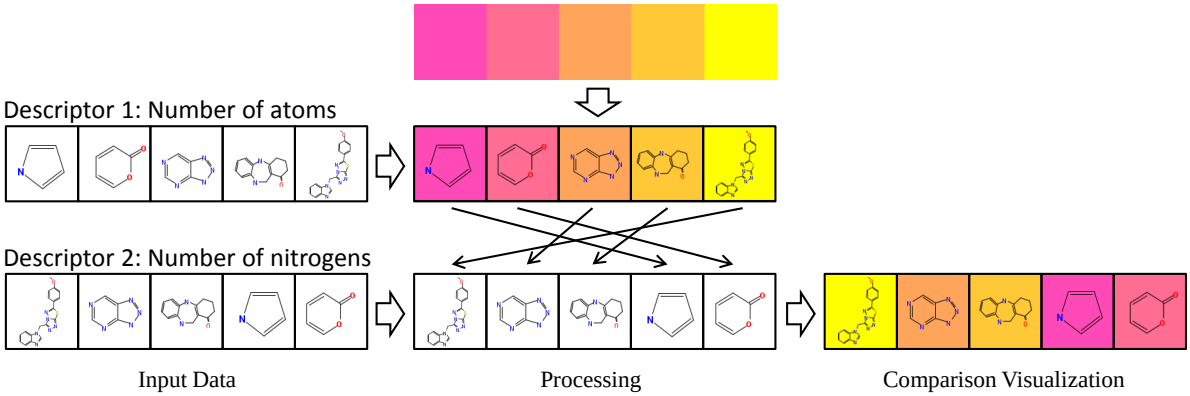
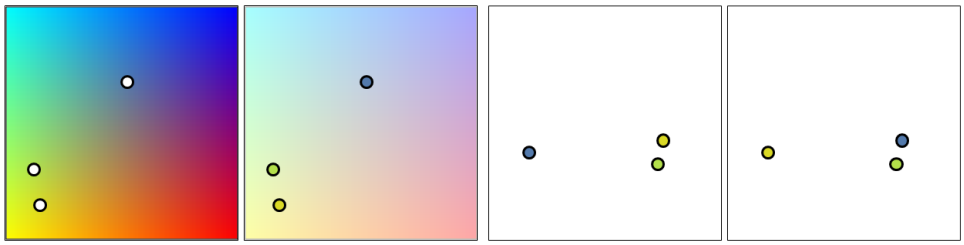


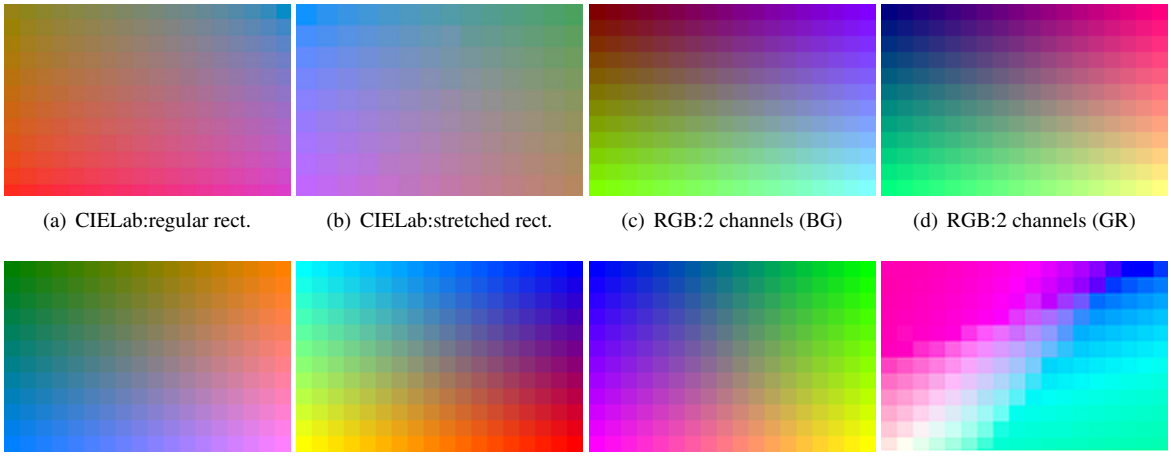
Figure 6.4.: Two meaningful data descriptors of biochemical data and their comparison. Each descriptor captures different data properties (atom resp. nitrogen count). Left: The input data is sorted according to each descriptor. Center: Color is mapped to the first ordering. The sorting is compared using connectors. Right: Compact comparison view. Color mapping based on object identity revealing descriptor correspondence.



(a) Reference coloring (b) Comparison coloring

Figure 6.5.: This image demonstrates the comparison of two plots. a) Reference plot. Objects are colored according to the reference 2D colormap. b) Two comparison plots, each representing a descriptor compared to the reference. In the first, the distance in color of the objects is similar to their distance in space, so it is regarded as similar to the reference. In the second comparison plot, color and space distance is not similar.

distance relations as good as possible. These approaches use extraction of subspaces from, for example, the RGB or the CIELab color space. Compared to RGB, the CIELab color space is perceptually linear, which is beneficial for expressing distance relations by color. In return, the RGB color space is a regular cube and therefore quite easy to implement, whereas the CIELab color space has an irregular 3D shape and suffers from an additional projection needed to access CIELab.



(a) CIELab:regular rect. (b) CIELab:stretched rect. (c) RGB:2 channels (BG) (d) RGB:2 channels (GR)

(e) RGB:2 channels (RB) (f) RGB:3 channels (g) RGB:3 channels (h) XYZ

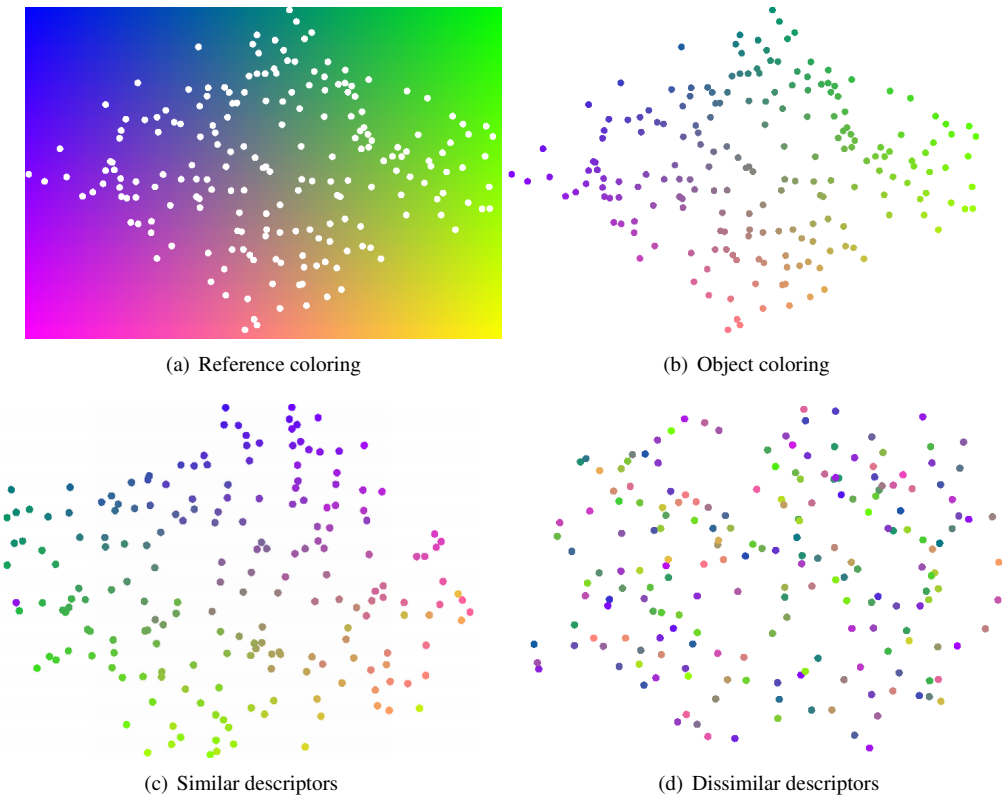
Figure 6.6.: Comparsion of colormaps for a 18x12 SOM grid. a) Rectangular cut out of the CIELab color space at L=55. b) Skewed rectangle cut (CIELab, L=55). c), d) and e) Two channels of RGB mapped to x and y axis, leaving the third constant. f) and g) Three channel RGB color scheme, diagonally cutting the RGB color cube [Him00]. h) Color scheme in XYZ color space.

The decision for a two dimensional color map is the result of a comprehensive comparison of current colormap techniques. Figure 6.6 contains some of the most promising color map approaches for the grid of 18x12 cells. Note that the grid resolution can be adjusted. In my opinion, the CIELabs, benefiting from a perceptually linear color scheme, cannot be adequately exploited, because either only a small linear subspace with a low color resolution can be used, or an additional nonlinear algorithm needs to be applied to project the SOM grid to the CIELab color space. A demonstration of the poor color resolution can be seen in Figure 6.6 a) and b). They show that the resulting color contrast is so low that adjacent grid coordinates can not be distinguished clearly. After extensive experiments with the RGB color space, considering two channel and three channel approaches, I made the decision to follow Himbergs approach [Him00] to use a linear section of the RGB cube with maximized color resolution. The goal was to increase the perception of color differences in SOM comparison. Thus, the colormap is spanned with the four corner colors cyan, yellow, blue and red (see Figure 6.6(g)).

6.2.4. Scatterplots

The scatterplot-based comparison of two descriptors relies on the display of individual objects. For comparison, both positions of one object in spaces D_a and D_b have to be shown in one plot. We color code every object in the projection of D_a using a 2D colormap (see Figure 6.7a,b). This color is assigned to the corresponding objects in projection p(D_b), respectively. If objects have similar neighboring objects in both projections, their neighbors have similar colors in the visualization. In this way, local and global similarities of the two compared

descriptors can be evaluated. In general, a homogeneous color distribution indicates a high similarity whereas a heterogeneous color distribution shows differences of the two descriptors (see Figure 6.7c,d).



(a) Reference coloring (b) Object coloring

(c) Similar descriptors (d) Dissimilar descriptors

Figure 6.7.: Scatterplot-based descriptor comparison visualization. Top: Object coloring. Bottom: The descriptor comparison. a) The reference color scheme mapped to the background and b) to the objects in the reference space. c) The homogeneous color gradient indicates a high similarity, d) the inhomogeneous gradient shows differing descriptors.

6.2.4.1. Projection Precision

Dimensionality reduction usually comes with a loss of information. In consequence, the 2D scatterplots contain errors, the relative positions of the objects in high and low dimensionality may differ. To reflect this, the projection error was calculated using the distance based quality measure which was also used in Section 6.2.1 [SvLB10]. The precision score can be mapped to the size, so that wrong positioned objects have an smaller influence on the perception of the visualization (see Figure 6.8).

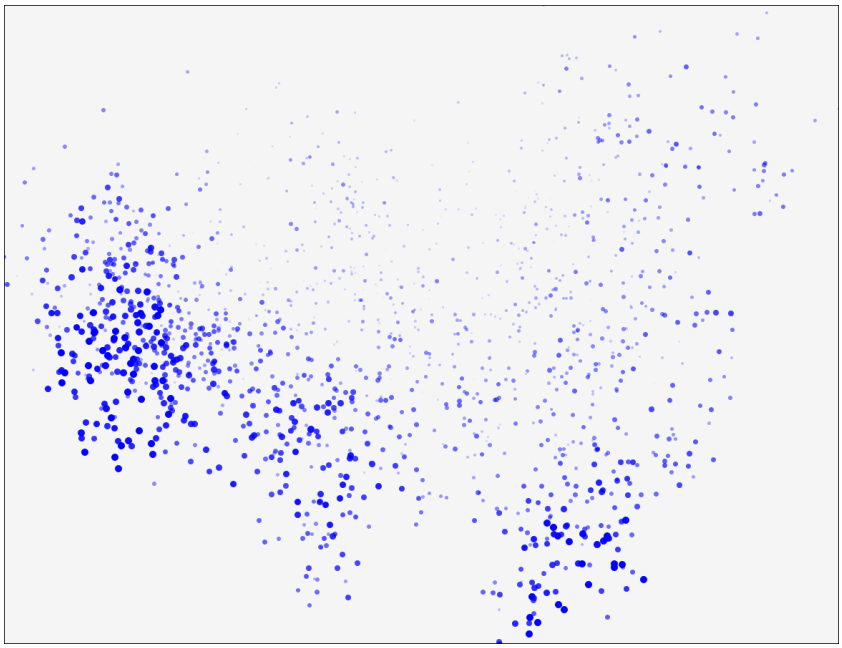


Figure 6.8.: The score of the precision of the projection is mapped to the size of each object.

6.2.5. Self-Organizing Maps

To ensure scalability, SOMs were used as they allow a good overview of the input space even for large data sets. They offer a 2D grid-based projection of an high dimensional space, where each cell can represent several objects. During the training process of the SOM, the neuronal network is dynamically adapted to the data, thus allowing a more effective representation of the data than a scatterplott with an static grid as overlay. In analogy to the scatterplot view, a two-dimensional colormap is used to indicate the neighborhood coherency. The color mapping is based on a coloring of the reference grid using a two-dimensional colormap (see Section 6.2.3 for details) and object correspondence between grids. In the compared grid, the color of each cell C_x^{comp} is determined by the color of an corresponding cell C_{corr}^{ref} in the reference grid. However, in SOM view, objects from one compared cell may belong to several cells in the reference grid. In my approach, C_{corr}^{ref} is determined by the position of the majority of objects $OC_x^{comp} \in C_x{}^{comp}$ in C^{ref} (see Figure 6.9 for an illustration). In Section 6.2.5.1, a further extension of this technique is presented regarding the reliability of comparison visualization.

For better visual quality of the SOM comparison view, the following additional data representations have been implemented: color interpolation for empty cells, color shifting for reference SOMs and visualization of color unreliability. All of this options are interactively steerable.

Depending on the input data structure, the result of the SOM algorithm can include few empty cells. They represent an area of the featurespace without data samples. The user can visually compare these areas in different feature spaces by interpolation of the colors of neighboring cells (see Figure 6.11(b)).

The SOM forms an adaptive grid, so the distances between neighboring cells are not constant. Therefore, I shift the colors of the two-dimensional colormap according to the distance of the SOM cell centers (see Figure 6.11c).

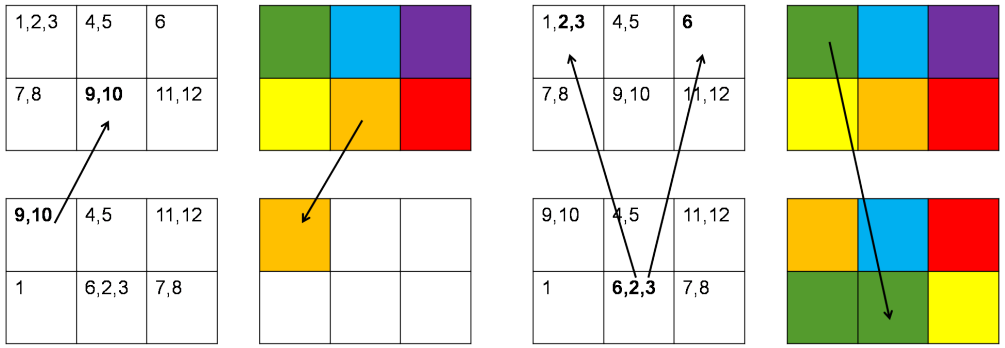
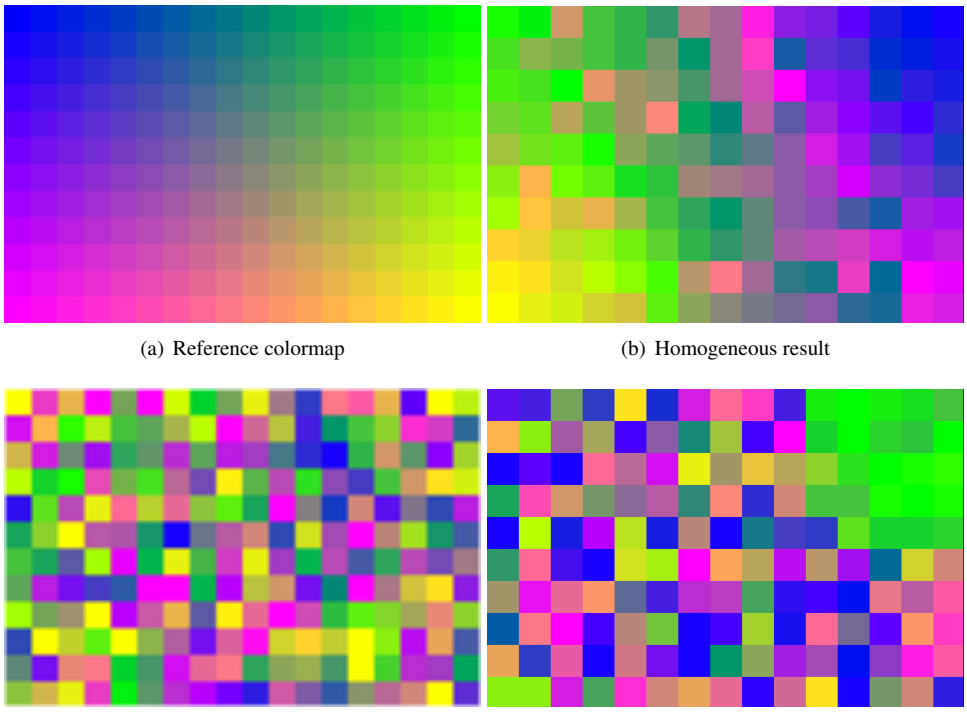


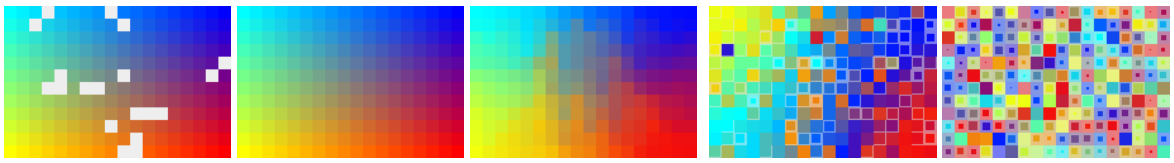
Figure 6.9.: Schema of the SOM comparison coloring. Top row: reference SOM. Bottom row: compared SOM. Left: An unambiguous color assignment, where all cell members from the compared SOM are grouped in one cell of the reference SOM. Right: The color assignment using majority principle – the cell color is used where the most elements are situated.



(a) Reference colormap (b) Homogeneous result

(c) Inhomogeneous result (d) Locally homog. result

Figure 6.10.: Grid-based descriptor comparison visualization using SOM. a) The reference color scheme, b) homogeneous color gradient indicating a high similarity, c) the inhomogeneous coloring for differing descriptors, d) locally homogeneous coloring showing descriptors well discriminating subgroups of objects.



(a) Original coloring (b) Interpolation (c) Color shifting (d) Coloring with reliability

Figure 6.11.: Visual extensions. Top: Illustration of improvement of visual display. Bottom: Display of coloring quality in SOM comparison. The columns show examples of SOMs. Left: a reference SOM, center: a homogeneous SOM, right: an inhomogeneous SOM.

This function resembles the so-called U-Matrix which helps to identify the structure of the SOM clustering [VA00].

6.2.5.1. Reliability

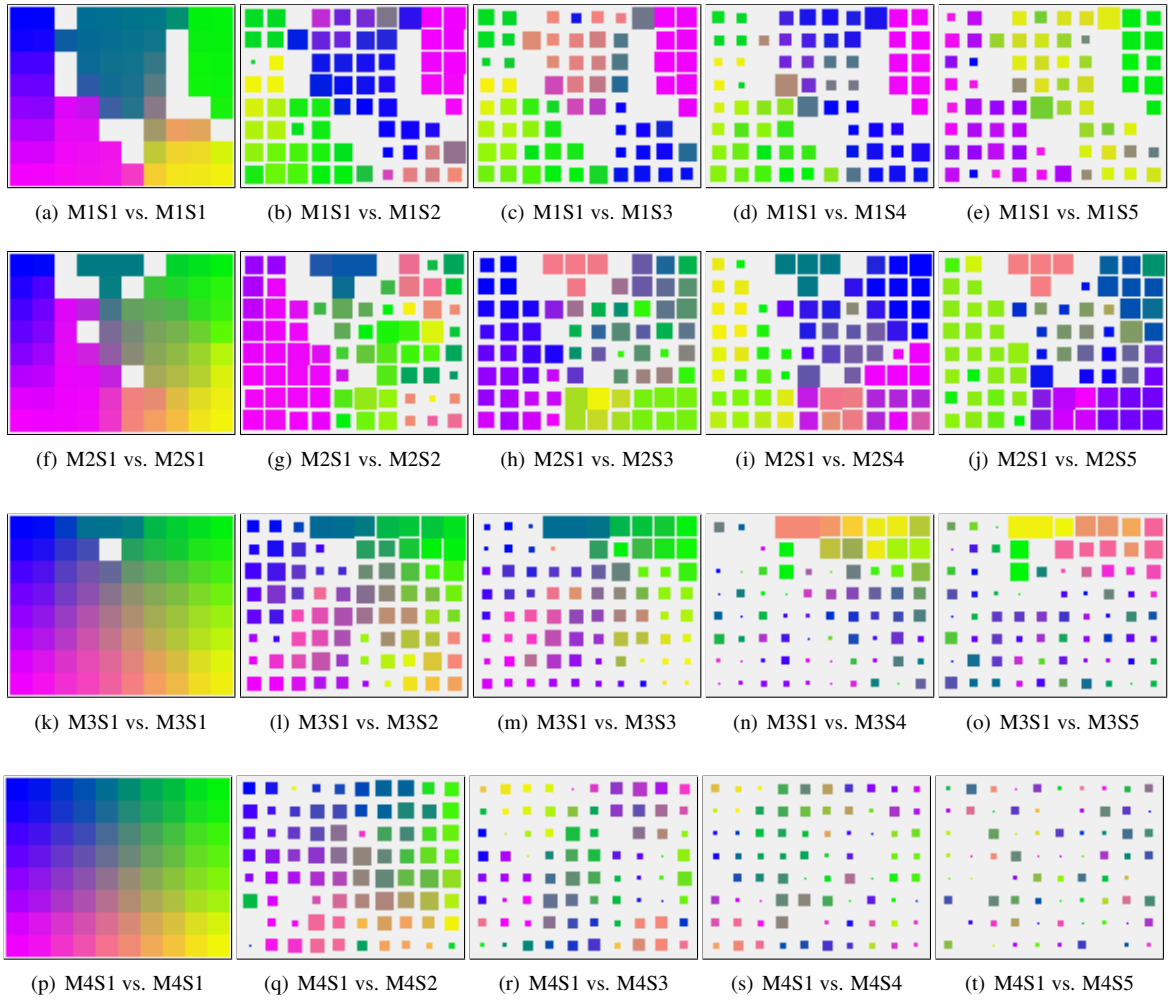
The SOM coloring uses matching of cell elements between the reference SOM and the compared SOM. In the easiest case, all objects O_i of the compared cell $|C_i|$ are in one cell of the reference SOM, so the unreliability of the cell coloring is zero (see Section 7.3.1 for details on SOM cell coloring). If the objects of a cell in the compared SOM (descriptor D_b) are distributed over several cells in the reference SOM (descriptor D_a), a higher unreliability is expected. The unreliability of a cell C_i in the compared SOM is measured by a score u_{C_i}, which takes into consideration the distance of the cells in the reference SOM to which the elements of the compared SOM cell C_i are matched and the selected majority cell R_i. The distance is calculated as the distance of the descriptors in the cell centers $d({R_i}^{D_a}, {R_k}^{D_b})$ using Euclidean distance measure. If the cells are located closely, the object distribution can be handled as similar.

As visual representation, unreliability can be mapped to the cell color (via alpha channel or one of the axes of the color space) or cell size (reduced corresponding to the unreliability score). Cell size encoding has turned out to be very effective and intuitive (see Figure 6.11d). The background color of the reduced cell is colored in the cell color with a higher, user steerable transparency. In this way, the impression of the SOM coloring remains stable, so the color gradient is still visible and on the other hand, the cell reliability is easy to evaluate.

6.2.5.2. Examples

Based on the synthetic dataset introduced in Section 6.2, different aspects of the visual comparison approach could be evaluated. For the comparison within the classes, the initial sampling of each model was compared with descriptors using the same model, but an increasing number of newly sampled objects (25%, 50%, 75% and 100%) (see Figure 6.12). As expected, the homogeneity of color decreases while the ambiguousness of the colormapping increases. Also this effect is visible in all models, it is accelerated in models with a lot of noise.

For the analysis of the comparison between different sampling models, the initial sampling of the first model was compared to the initial sampling of all other models (see Figure 6.13). Similar cluster structures are clearly visible just as the dissimilarity in the noise regions is explicit, too. Please note, that a class in one SOM does not necessarily have the same color as in an other SOM as SOMs a rotationally invariant.



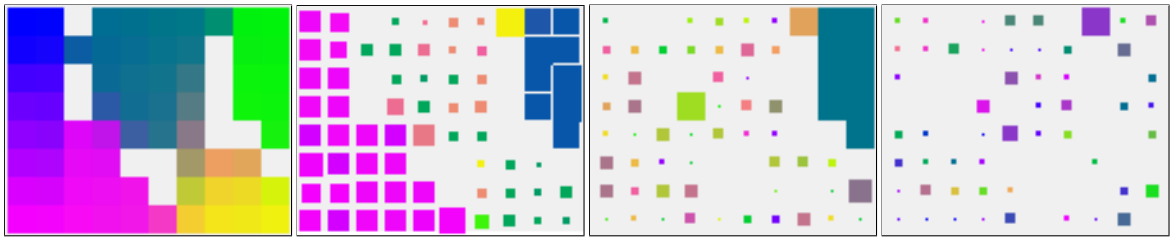
(a) M1S1 vs. M1S1 (b) M1S1 vs. M1S2 (c) M1S1 vs. M1S3 (d) M1S1 vs. M1S4 (e) M1S1 vs. M1S5

(f) M2S1 vs. M2S1 (g) M2S1 vs. M2S2 (h) M2S1 vs. M2S3 (i) M2S1 vs. M2S4 (j) M2S1 vs. M2S5

(k) M3S1 vs. M3S1 (l) M3S1 vs. M3S2 (m) M3S1 vs. M3S3 (n) M3S1 vs. M3S4 (o) M3S1 vs. M3S5

(p) M4S1 vs. M4S1 (q) M4S1 vs. M4S2 (r) M4S1 vs. M4S3 (s) M4S1 vs. M4S4 (t) M4S1 vs. M4S5

Figure 6.12.: Visualization of the comparison of the sampled data within each sampling model (see Table 6.2). Each row, representing one model, starts with a comparison of the initial clustering to itself. The following columns contain instances of the same model with a resampling rate of 25%, 50%, 75% and 100%. The homogeneity of the color gradient decreases with increasing resampling rate.



(a) M1S1 vs. M1S1 (b) M1S1 vs. M2S1 (c) M1S1 vs. M3S1 (d) M1S1 vs. M4S1

Figure 6.13.: Comparison of the different sampling models to model M1. The white spaces in the SOMs represents empty cells which contain no objects. The homogeneity of the visible color gradient decreases with the decreasing similarity of the artificial ground truth data models.

7. Visual Descriptor Comparison Pipeline

7.1. Introduction

The previous chapters presented techniques for the analysis of hierarchical structured datasets and for datasets with an unknown structure. Although the used methods differ in their respective realization, the concepts have some profound commonalities. Requirements and expectations on the exploratory analysis of descriptors vary from task to task and from user to user. In some cases, only a fast grouping and calculation of representative descriptors is desired. In other scenarios, the user needs to „dig deeper“. Either because the automatically calculated output was not the expected or because he wants to understand the underlying characteristics of the data or the used algorithms and their parameters to gain more insight. As an analysis process, especially when dealing with large datasets, is complex, an interactive pipeline can help to guide the user through this process. Usually, in such a pipeline the users starts with the first steps (e.g. steering some initial parameters) and work their way up to the final calculation results. This approach works well if the users are familiar with the process, the used data mining algorithms, their parameters and a well defined analytical goal. Untrained analysts have to juggle through the pipeline or just hit the default option.

In this Chapter, a generic pipeline for the visual comparison of descriptors is presented which can be applied to both approaches. In the propose pipeline the user interaction starts in the last step of the calculation, the result view. The analysts get a proposed result of representative descriptors and it is up to them to work with the result as it is or take a closer look at the analysis process.

The work presented in this chapter has been partially published in [BvLBS11].

7.2. Related Work

Finding relevant descriptors as low dimensional representation is closely related to my work and can be a part of an exploratory descriptor analysis. Hence, I will present some relevant work of interactive systems for dimensionality reduction. As already mentioned, data descriptors consist of a set of features (numeric values) representing complex data types. However, finding a proper descriptor is a challenge. Low dimensionality may lead to under-representation of the objects, and high-dimensional descriptors may suffer from problems such as “curse of dimensionality”, where the distances between near and far objects converge [AHK01, BGRS99].

The usage of algorithmic approaches solely is difficult for non-expert users as they need setting of, possibly extensive, number of input parameters and work in a black-box manner.

Integration of user feedback in the analysis process is crucial when use case dependent parameter adjustment and result evaluation are required. Many approaches interactively combining automatic calculations with visualization in the area of feature selection have been proposed. Choo *et al.* [CLKP10] presented a framework for data classification combining matrixes with parallel coordinates. A scatterplot shows the resulting projection of a Linear Discriminant Analysis [Fuk90] which is iteratively refined during the classification process. Usually, dimension reduction methods focus on preserving structures of the high dimensional space. Johansson

and Johansson enable the user to rank the importance of those structures by interactive steering of quality metrics [JJ09]. Tatu et al. [TAE*09] proposed analytical methods to find and filter important structures to reduce the complexity of the resulting visualization. The DimStiller [IMI*10] framework supports the whole process of feature selection. Additionally, the user gets guidance in every step of the pipeline, e.g., regarding parameter choice. It however focuses on dimension reduction only for individual features and does not consider local similarities. Building upon these approaches, I propose a strategy where the user can decide on the level of process automation from a fully automatic up to a step by step assisted work flow.

7.3. Approach: Guided Interactive Analysis Pipeline

My approach aims at guiding the user through the process of finding representative descriptors of a given dataset. Each of the selected descriptors should reflecting other aspects of the data. The input consists of several multivariate descriptors, each potentially having a different dimensionality or different data type. The output is a subset of independent descriptors suitable for the use case at hand. In the workflow, the descriptors are compared pairwise for finding groups of similar ones and thereby to choose representatives among them.

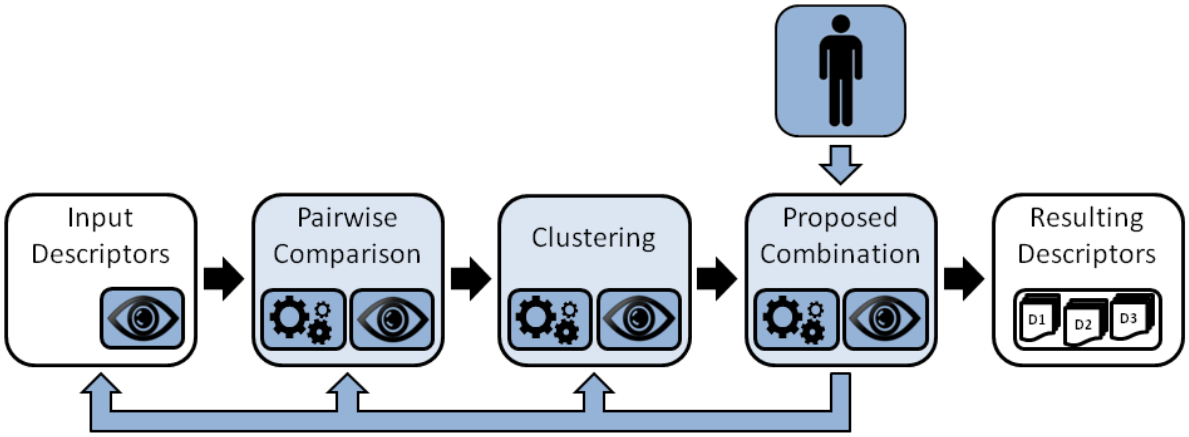


Figure 7.1.: Schema of the descriptor selection process. Every step (blue) encompasses automatic data processing and visualization part. The input consists of many descriptors for one data set. These are automatically compared and filtered. The result is a proposed set of independent descriptors is the starting point for the interactive, guided analysis process. Feedback loops allow the user to refine results on demand.

In order to adapt the algorithms to a given use case and improve result specificity, the user is involved in the process. Offered guidance should facilitate the analysis for non-expert users. I have developed a dedicated pipeline supporting the data descriptor finding process (see Figure 7.1). This pipeline consists of several steps that are supported by both algorithmic and visual means. Every step of the pipeline supports visualization and is interactively steerable. By combining the automatic and visualization functions, the results can be iteratively refined by the analyst.

First, the input data (a set of descriptors) can be explored using dedicated visualizations. They support the scalability with respect to the number of input data items. For large data sets, visual clustering using SOM is employed. Then, a pairwise comparison of the descriptors shows both, global and local similarities between them.

These results are used for filtering of similar descriptors. The final recommendation step shows an overview of results of automatic pre-processing, recommendations and offers the user the possibility to interactively refine the results. The analyst receives additional guidance by the highlighting of interesting or critical patterns. Feedback loops allow the user to interactively refine the results.

7.3.1. Pairwise Descriptor Comparison

The first step of the pipeline is the pairwise comparison of all input descriptors. Based on this, groups of similar descriptors can be formed and redundant information can be found. Depending on the given input data, different techniques can be applied. In this work, I propose measures for trees and high dimensional descriptors with varying dimensionality. In other scenarios, the focus could be shifted to e.g. correlation or class label based approaches. The automatic calculations should be accompanied by interactive visualizations supporting the user. These should emphasize the key tasks in the comparison process like the identification of local dissimilarities in globally similar descriptors. I present examples for comparative visualizations based on tree visualizations (see Section 4.2.4 and 5.2.1), on scatterplotts (see Section 6.2.4) and on SOMs (see Section 6.2.5);

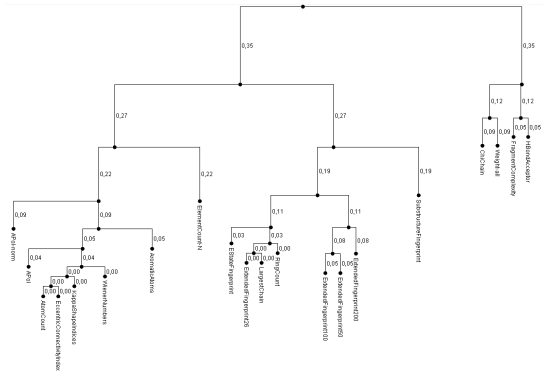
7.3.2. Hierarchical Clustering of Descriptors

Finding patterns in the descriptor space is one of the goals in their exploratory analysis. Due to the pairwise comparison, distances between the representations are available and can be used for clustering. I use a hierarchical clustering approach (average linkage) to structure the data. This approach is beneficial as no additional parameters except the linkage mode are required a priori. The resulting tree is visualized whereas the distances are mapped to the edge lengths. The user can now interactively set a threshold for the maximum edge length or select a number of clusters to determine the final clusters. Alternatively, the number of clusters can be estimated automatically using the knee/elbow criterion for different quality criteria [TW05, TWH01, SC04]. This can be used to find an appropriate number of clusters in combination with various clustering algorithms [SC04], but in combination with hierarchical clustering it is not associated with computational expense. The respective linkage-distance in each step serves as quality criterion for a cluster, the knee/elbow is marked by the point with the biggest difference between two steps [CFC*01].

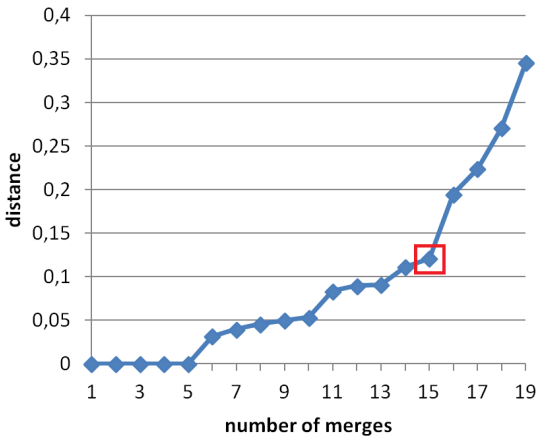
This process not only gives the user more insight, it additionally can be used to reduce the complexity of the descriptor space. High similarity of two or more descriptors implies that they carry the same information regarding the neighborhood distribution of the described objects. A representative descriptor can be selected, removing redundant information and reducing the dimensionality of the descriptor space. In my approach, the one with the minimum average distance to all descriptors within the same cluster is selected. Note that the pre-filtered descriptors can be interactively viewed and the pre-filtering can be rejected by the user in the next step of the pipeline. This is in particular important for borderline decision cases or in cases where user knowledge contributes to the decision making.

7.3.3. Recommendation Visualization and Exploration

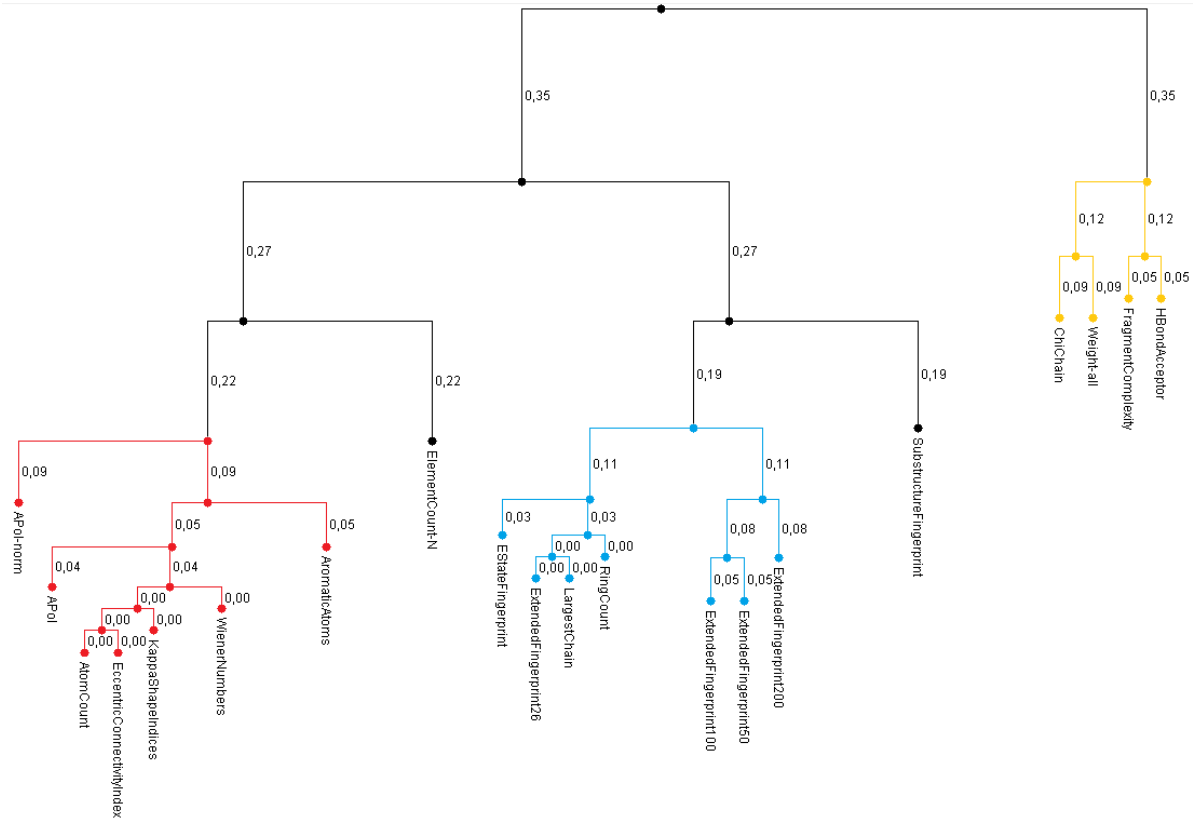
The automatically calculated proposal for the compact representation of the descriptor space is presented to the user in the last step of the pipeline. The result inspection is supported by interactive visual exploration of descriptor similarity. The user can choose from just applying the proposed combination or to inspect and adjust the steps of the process. User involvement in the process is advantageous especially in borderline cases where the automatic filtering decision is close to the decision criteria (similarity threshold). Interactive data



(a) initial clustering



(b) cluster distances



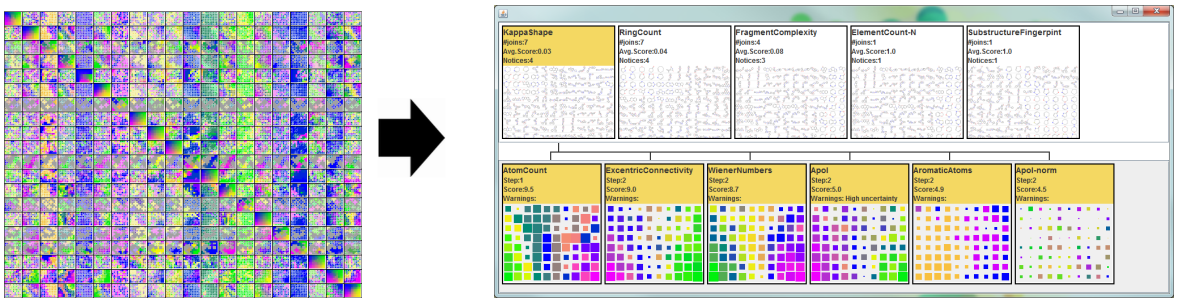
(c) hierarchical clustering with proposed clusters

Figure 7.2.: This figure shows the result of a hierarchical clustering of 20 input descriptors (a). The edge length reflects the distance of the descriptors. The threshold distance to define the highlighted clusters was automatically determined using the elbow/knee criterion (b). The resulting clusters are individually colored (c).

space exploration can reveal new information, helping the user to make better decisions on descriptor filtering. For example, the scoring function cannot reveal local similarities between objects in two descriptor spaces. However, these can be highly relevant for the usage of descriptors for certain object classes (e.g., humans in 3D object databases). The inspection is supported also in algorithmic way, highlighting such borderline decision cases. The algorithmic and visual support of the adjustment process is described in the following.

During the clustering process, patterns which are recommended for further examination by the analysts are automatically detected. These so-called examination markers can e.g. be cases where the distance to the next cluster was very close to the threshold. This may indicate low confidence of the automatic cluster assignment decision

The recommended set of descriptors is visualized on the basis of their two-dimensional projections (see Figure 7.3b top row). The view includes additional important descriptor information (e.g., the number of similar descriptors or examination markers). This summary overview is an entry point to a deeper examination of the decision space. For example, the comparison of one selected descriptor with similar filtered descriptors is shown on demand (see Figure 7.3b bottom row). In this way, the understanding and adjustments of the filtering results are supported. This exploration may lead to adjustments in the process – feedback to the previous interactive steps of the pipeline.



(a) Descriptor comparison matrix

(b) Result view

Figure 7.3.: Visual descriptor comparison. a) Initial overview of pairwise descriptor correspondence. b) The result view after descriptor filtering. The top row shows the representative of the clusters or unclustered descriptors. The bottom row shows the comparisons of one descriptor (top row in yellow) with related descriptors of the same cluster. This supports understanding of the filtering decisions.

7.4. Application to Pharmaceutical Data

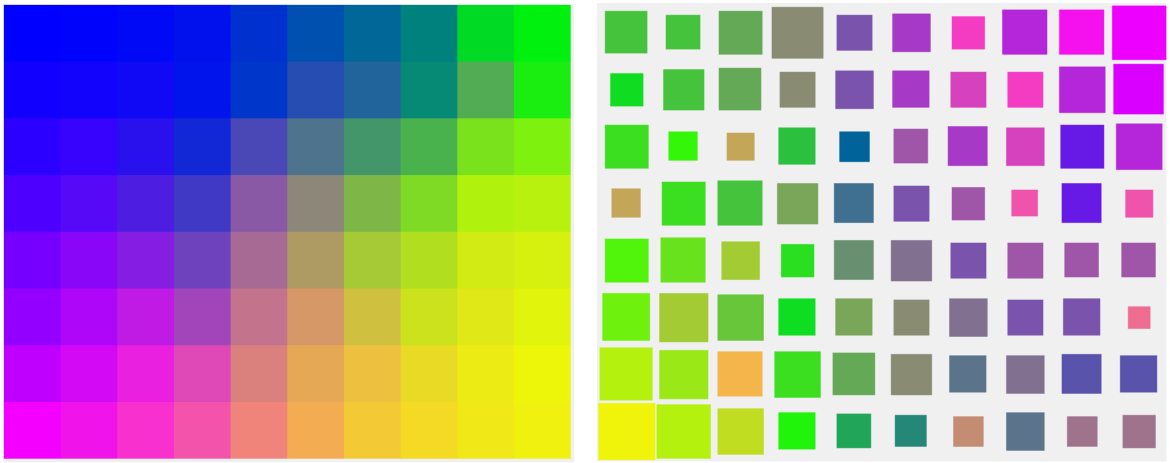
In this section, the approach is demonstrated on a biochemical dataset following the introduced workflow. Researchers in biological and pharmacological sciences analyze large sets of molecules, e.g., as output of High Throughput Screenings (HTS). In HTS, many molecules are tested for reactivity with one specific molecule of interest. The resulting datasets contain several hundreds or thousands of molecules with high reactivity. The task of the analysts is to find few, promising compounds for further examination. The selection criteria are use case depended. Not only the data structure, but also user expertise and further factors such as costs have to be considered. Moreover, as shown in Figure 6.4, often there is more than one valid description of a given dataset. Therefore user interaction in the analysis process is needed.

The dataset contains 9989 molecules, described by 20 standard pharmacophore descriptors, divided into two groups. The first group consists descriptors of 13 basic, Quantitative Structure-Activity Relationships (QSAR) for, e.g., fragment complexity or the number of hydrogen-bond donors with a dimensionality between one and ten [BST04]. The second is a set of so called *fingerprints*, binary descriptors classifying whether the described molecule fulfills certain conditions or not. The group consists of 7 fingerprints with a dimensionality from 26 to 400.

The analysis task combines two intentions: 1) Finding relevant groups of compounds and structures in the data, and 2) find differences and commonalities to reveal the underlying characteristics of the data. The identification of groups is important for the process of drug development. Single molecules with high activity might be outlier and form no reliable basis for further (expensive) research. Larger groups of active molecules, especially if they share a common backbone, are more promising candidates. Their compact description ideally captures only the relevant chemical features which caused their reaction characteristics. Based on this, new active compounds which were not part of the HTS analysis can be extrapolated.

As outcome of the automatic analysis process, three different groups and two single descriptors were proposed. Their overview is shown in Figure 7.5. One of the larger groups containing seven descriptors is represented by the KappaShape descriptor. The quick comparison visualization in the result view already offers more insights (see Figure 7.3). The number of different molecule groups differs e.g., the AtomCount descriptor has a homogeneous color gradient showing around 5 different colors (groups) whereas the AromaticActoms descriptor shows only one big and two smaller groups. The polarization descriptor (Apol-norm) has the lowest similarity to the reference, which is reflected by the low reliability of the cell correspondence.

To get more details, the analyst takes a closer look on the previous clustering step (see Figure 7.2) of the analysis pipeline. It illustrates the position of the polarization descriptor at the border of the red cluster. Some of the descriptors have only very small distances, which is also visible in the direct comparison (see Figure 7.4). The



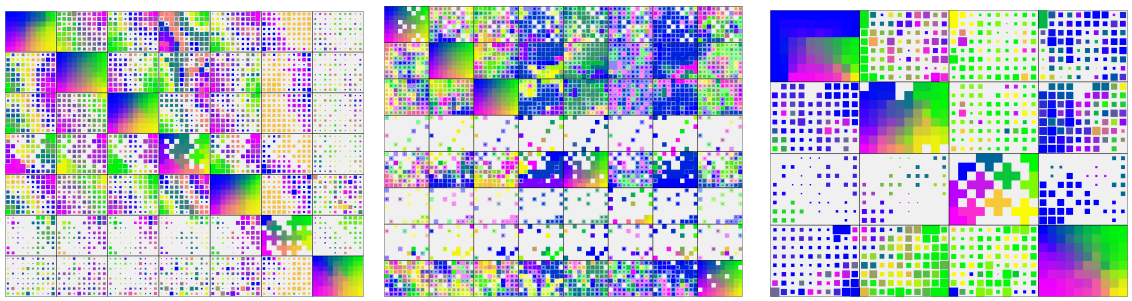
(a) WienerNumber (b) WienerNumber vs. KappaShapeIndices

Figure 7.4.: The WienerNumber and KappaShapeIndices descriptors have a very similar topology of the described molecules. This is also visible in the comparison visualization, showing a very homogeneous color gradient.

blue cluster is is dominated by the fingerprint descriptors. In this dataset, most of the complex, high dimensional

fingerprints could be represented by the only one dimensional ring count. This finding is not only chemically relevant but helps to reduce the dimensionality for the representation of the dataset in further data mining or simulation steps.

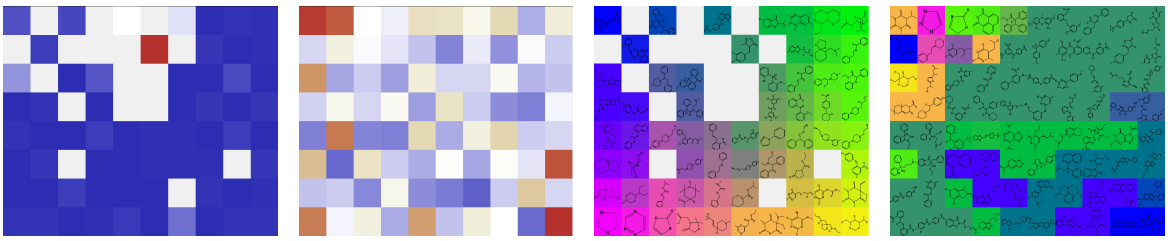
Going one step back again in the pipeline brings the analyst to the comparison matrix (see Figure 7.5). Some of the SOMs e.g., the three very similar descriptors LargestChain, RingCount and ExtendedFingerprint26 have only few populated cells in the corresponding SOMs. Some of the SOMs are dominated by one color (e.g.



(a) red group (b) blue group (c) green group

Figure 7.5.: Comparison matrices of the three clusters.

ExtendedFingerpint50 and ExtendedFingerpint100) (see Figure 7.6). A detailed inspection reveals, that most of the elements in the ExtendedFingerpint50 are concentrated in only one cell. Here, the increase of dimensionality of the ExtendedFingerpint100 is useful it helps to distribute especially these compounds over the SOM. Although there is a big overall distance between the groups, the visualization helps to find local similarities e.g., between the APol and the ExtendedFingerpint100 descriptor.



(a) FingerPrint50 (b) FingerPrint100 (c) FingerPrint50 (d) FingerPrint100

Figure 7.6.: a) Most of the molecules are concentrated in one cell of the FingerPrint50 SOM. b) The FingerPrint100 SOM shows a better distribution. c) Shows the reference coloring applied to the FingerPrint50 SOM. d) The comparison visualization reveals, that the FingerPrint100 SOM especially distributes the objects witch are agglomerate in the FingerPrint50 SOM. In this case, the additional 50 dimensions of the FingerPrint100 help to enhance the analysis.

7.4.0.1. Summary

In this exemplary analysis scenario, each step revealed more information. The analyst could work with the proposed result only, giving information about the number of clusters and the respective dominating descriptors and their chemical properties. Digging deeper helps to find interesting local characteristics and anomalies, which might lead to a better understanding of the data and refinement of the analytical methods e.g., choosing other threshold parameters or calculating additional descriptors.

Part II.

Exploration of Time Dependent Data

8. Introduction

Time dependent data appear in many scientific and business areas like climate or wildlife research, stock trades or clinical test series. Time series and their analysis is a broad field, but in general, one or more properties of the observed objects are measured at certain time points. This could be, for example, the position in longitude and latitude of cars and their actual movement speed at every given point in time. The connection of consecutive points in time to descriptors places high demands on the approaches which are applied for the exploratory analysis. A broad variety of Visual Analytics solutions was proposed and are specialized on the analysis of time series.

Most often, visualization of time series focuses on one dimensional time series, where the different properties of the observed objects at each time step are observed independently. This makes it difficult to find patterns involving additional information like dependencies of features or behavior of groups of objects. If the analyzed time series is very long, not every point in time can be displayed simultaneously. In these cases, either some moments are selected for visualization or several time points and their properties are merged. Each of these techniques is represented in my work. For the analysis of categorical, time depended data, I combine automatic selection of interesting time points with an interactive visualization called Time Parallel Sets (TIPS) (see Chapter 9). For the exploration of sequenced time series, where each symbol represents similar input subsequences, I combine different linked visualizations, each specialized on other aspects of the data (see Chapter 10).

9. Categorical Time Dependent Data

In this chapter, the analysis of time-dependent data of grouped or categorized objects is addressed, which is relevant to various application domains. The groupings may be pre-defined (e.g., animal herds in biology), or may be a result of previous analysis (e.g., clustering). In both cases, when the group membership changes over time, it is necessary to examine these aspects (e.g., which herds change members and when).

The analysis tasks include:

- tracking group sizes and contents over time,
- tracking (sub)sets of objects (e.g., a set of objects changing the same groups between two points in time),
- tracking group memberships of individuals,
- analyzing group changes (how many group members switch to which group),
- examining objects added to and removed from groups (e.g, newly born or died animals).

The main challenge is the scalability with respect to the number of group changes and the number of time points. Therefore, a visualization combined with automatic data analysis is needed. The related works either only focus on interactive visualization of group changes [KBH06,ZKG09,LSP*10], or only on time development of the group sizes [HHN02].

To ensure scalability, I present Time Parallel Sets (TIPS), a detailed visualization of interesting time points and group dynamics combined with an automated time series analysis. I propose different algorithms for the calculation and selection of interesting points in time, depending on different group dynamic patterns.

The approach is demonstrated in two application areas: Emergency evacuation of buildings and activity tracking of people over long time periods. An application to weather data can be found in [vLBA*12]. The application cases were analyzed in cooperation with Natalia and Gennady Andrienko from Fraunhofer IAIS, Bonn.

The work presented in this chapter is partially based on [BvLAA11] and [vLBA*12].

9.1. Related Work

This work relates to the areas of spatio-temporal visual data analysis, visualization of (time-varying) categorical data and analysis of time series. As dynamic categorical data often stem from clustering of other time-varying data, an overview of the work in this domain is given as well.

Visualization of categorical data: The Parallel Sets approach by Kosara et al. [KBH06] displays multivariate categorical data. In particular, it shows group memberships across several categorizations building upon parallel coordinates [Ins85]. Similarly to Parallel Sets, Interactive Sankey diagrams [Fro05] show categorical data over several dimensions while improving the visual design. Mosaic plots [Hof08] and KVMap [May07] use space filling approach for multi-dimensional categoric data. Mosaic plots divide the rectangular space recursively according to number of objects in each category. KVMaps use regular splitting, where color coding shows the number of objects in each category.
Categoric data in geographic context is often shown on maps (e.g., by color coding of geographic areas according

to category value) or using special visualization techniques. For example, for hierarchically organized categoric data, spatially-ordered treemap technique has been introduced [WD08]. Recently, Wood et al. [WBDS11] presented BallotMaps – an abstract visualization of spatial ordered data (voting rankings for voting regions). It shows the relative preference of a candidate for regions and for parties (categories) according to name order. All of these approaches however do not deal with categories changing over time.

A special case of categorical data are clustering results. Zhou et al. [ZKG09] and Lex et al. [LSP*10] propose approaches for **clustering result comparison**. They both consider several groupings of data objects according to different clustering results. The visualization is similar to Parallel Sets [KBH06], however it focuses on individual objects. The view connects each object across all clusterings while minimizing edge crossings. These approaches do not regard time dependency of the data and put strong emphasis on individuals rather then groups.

Visualization of time-varying data: A recent book on visualization of time series [AMST11], built upon the survey presented by Aigner et al. [AMM*08], provides a broad overview of techniques for visual analysis of time series. Most relevant to my work is the approach of Hao et al. [HDKS07], who proposes an importance-driven time series visualization. In the view, more important time intervals are provided with more screen space. Data which are more up-to-date or have higher variance are deemed more important. Moreover, Ziegler et al. [ZNK08] visualize time series based on the changes between pairs of time steps and in this way, important value changes can be identified.

Visualization of time-varying categoric data: ThemeRiver [HHN02] approach shows time changes of topic categories over times in a stacked chart, where bar sizes denote the number of objects in each group (i.e., number of articles per topic). The developments in the number of objects in each category can be examined. A similar approach is the History Flow [VWD04], which displays the development of documents by authors. The focus is on identification of important themes/authorships in each time step. Both approaches, however, do not include the changes of group membership between time steps.
Sequence data also can be seen as time-varying categorical data without an explicit time dimension. There are several approaches for visual analysis of such data. They disregard the exact time and concentrate only on states (sequence values) and their changes. Often they are represented as state-change or state transition graphs, where each node is a state and an edge represents a state change. Edge thickness and node size represent the number of objects [vHvdWvW01, VJC09, BBG*09]. The main disadvantage is that the exact time of transitions is not visible in the graph. It can only be seen in another linked view. Moreover, these approaches do not take spatial information into account.
Recently, a new approache for analyzing group changes over time have been presented by Turkay et al. [TPRH11], which deals with clustering results of time-varying data. The visual analysis shows which cluster an individual object belongs to over time (similar to [LSP*10]) and focuses on assessment of cluster quality development. They however, disregard geographic location of the data, so spatio-temporal analysis is not possible and the selection of points in time, which is crucial for large datasets, is not provided.

Spatio-temporal visual data analysis: The paper by Guo et al. [GCML06] demonstrates two approaches to exploration of categorical changes that are applied in geovisualization. In this work, categories of spatial objects (states of the USA) are defined by clustering of combinations of values of multiple time-varying attributes. The authors suggested two complementary ways to visualize the cluster membership of the objects over time. The first is the classical “small multiples” approach [TH83], where a small map for each time step in which the cluster membership of the objects is represented by color. The second is a matrix with the rows corresponding to objects and columns to time steps; the cells are colored according to the cluster membership. The ”small multiples” are good for exploring the spatial distribution of the categories at each time step and the differences between the distributions at different time steps. The matrix is suitable for exploring the evolution of each individual object. However, both displays do not support the exploration of object groupings and detection of coherent changes of

multiple objects sufficiently. Besides, it is cognitively difficult to examine a large number of maps showing data at different time steps. The displays also do not provide a convenient overview of the overall evolution.

Automatic identification of significant time steps in time series data: The above-mentioned approaches to visualization of categorical data cannot be straightforwardly applied to long time series of categorical values. There may be not enough screen space to represent all time steps and changes between them. Hence, there is a need for scalability in the approaches, e.g., representing all time steps in a summarized form and selected time steps in more detail. Good selection of time steps plays a crucial role.

There are numerous visual and computational methods for identification of important time steps in time series [HJM*11, ZJGK10, ZNK08, LH11], videos [PDT09, TV07] or other types of data. They try to compress the underlying data set into a small set of key time steps that would best represent the data set. There are various methods, such as largest difference between time steps, regular spaced time steps with succeeding removing of less important time steps. They mostly work with quantitative data, so they need to be extended or adapted to categorical data.

9.2. Approach: Time Parallel Sets (TIPS)

The proposed data display called Time Parallel Sets (TIPS) extends the Parallel Sets concept [KBH06] with the time dimension and includes additional glyphs for easier identification of group changes. I introduce:

- The tracking of group changes on three levels: whole groups, sets of objects and individuals.
- The automatic data analysis selecting representative points in time for deeper examination. The automatic identification of interesting time points focuses on patterns of group changes of objects over time. It supports different interestingness functions and a visual feedback to find appropriate parameters.

9.2.1. Interactive Visualization

The visualization shows two levels of detail (see Figure 9.1). The groups are color-coded and shown in user-defined order on vertical axis. On the bottom, the distribution of all objects among groups at every moment is shown. The main view in the upper part displays the object distribution (bars) and the group changes of objects (edges between bars) in selected time moments, whose temporal positions are shown in the lower part by vertical lines.

In the main view, each vertical bar represents the group distribution in one moment. Edges between neighboring bars represent objects group changes between two points in time. Their color is optionally based on source or destination group or has a color-transition from one to the other color. The changes can additionally be shown by the color coding of the narrower bars aside of every group. It indicates the proportional group membership of the contained objects in the time steps before and after the current one. It is especially useful when the outgoing/incoming edges are strongly overlapping or to increase space efficiency, if many points in time are displayed. The dataset analyzed does not necessarily contain the same / all objects in every point in time. For representing this fact, added and removed objects are indicated as triangles at the side of the group bars (see Figure 9.2).

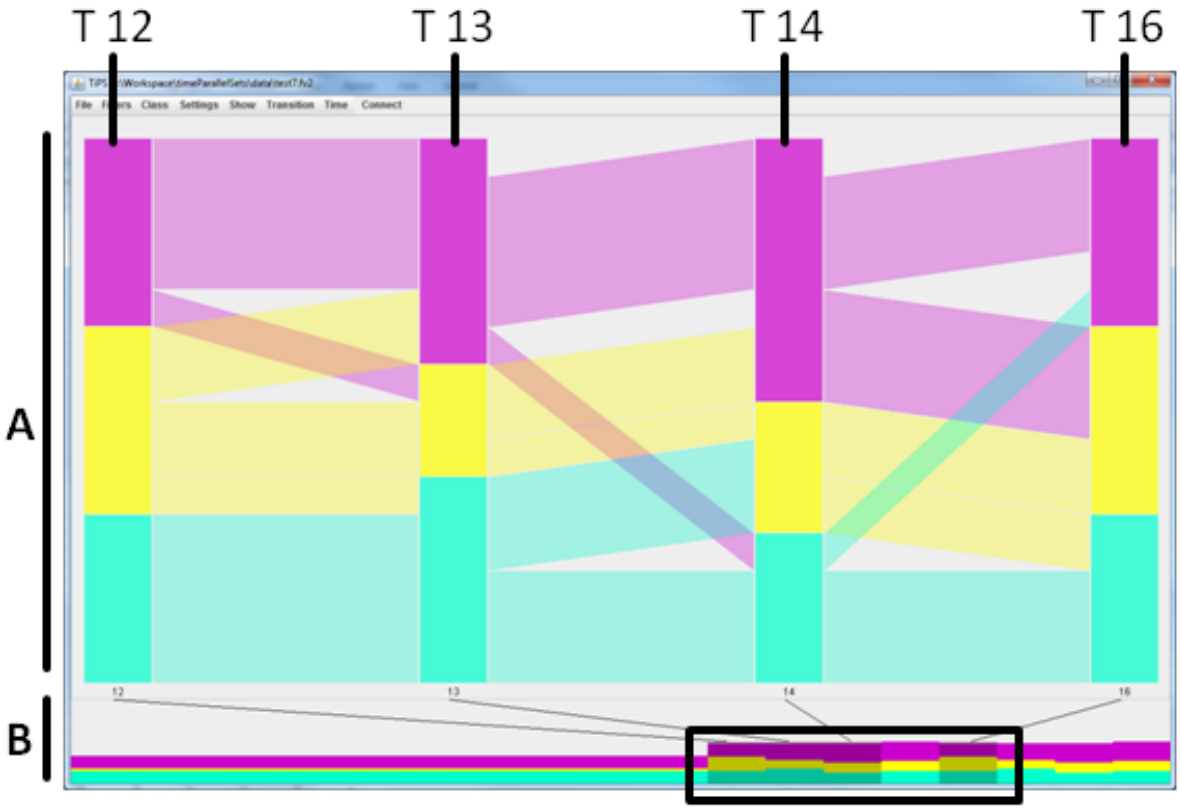
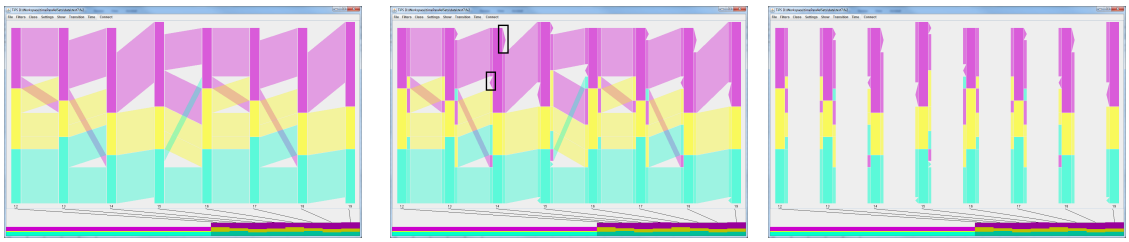


Figure 9.1.: Exemplary TIPS visualization. The Display is divided into two parts. Each group is represented by one color. A) The upper part shows the group distribution (bars) and group changes (edges) in selected time points (T12, T13, T14 and T16 in this case) B) The lower part shows the object distribution in the whole time period. Here, time points of interest can be selected and then are highlighted in gray (black box)

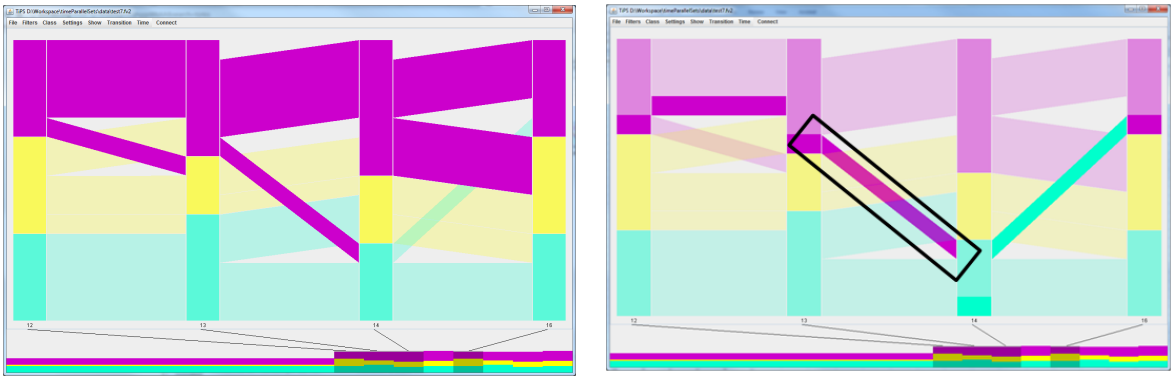


(a) (b) (c)

Figure 9.2.: Example visualization of box-border properties. a) Visualization of eight points in time without box-borders. b) The borders on each box reflect the target of the edge. Object which are not present in the previous or following time point are indicated by triangles (black box). c) Visualization without edges to increase space efficiency.

To enhance the data analysis, not only the overview upon group changes but also the tracking of sets of interesting objects over time is important. Tracking encompasses three different aspects of the data which are highlighted:

1. *Tracking of groups:* The bars of the selected groups are highlighted as well as all leaving edges. The user can comprehend the characteristics of this group and the objects contained, e.g., if leaving objects always swap together to one group or not (see Figure. 9.3(a)).
2. *Tracking group changes:* A transition matrix shows the number of group changes aggregated over time as a heat map. The cells indicate common and extraordinary group changes. They are linked to the main view for highlighting or hiding the represented transitions. Thereby, rare transition can be identified and analyzed in more detail. (see Figure 9.4).
3. *Tracking of individuals and object sets*: An object or a set of objects can be selected, e.g., by clicking on an edge. Consequently, all bars, side bars and edges containing the selected objects are automatically highlighted proportional to group size. In this way, the transitions of a set of objects across groups can by examined, e.g., to check if the set objects have common transitions (see Figure. 9.3(b)).



(a) Selection of group

(b) Selection of transition

Figure 9.3.: Example visualization and tracking. a) The purple group was selected. All purple bars and edges are highlighted. b) A transition (black box) was selected. The contained objects are highlighted in all other groups or transitions.

9.2.2. Automatic Selection of Intersting Time Points

The analysis of categorical changes (i.e., transitions), as proposed in this work, relies on a adequate selection of time points. Regarding the limited screen space and cognitive human skills, it is not possible to visualize every step in the main view. The more compact overview visualization in the bottom window allows a visualization of the fraction of each class in every point in time, but important group changes might be hidden (see Figure 9.5). A constant or random sampling of time points might miss relevant information, too. I propose to present only those time moments (and categorical changes) to the user, that uncover important data properties. I have therefore developed several algorithms for user-guided automatic time selection. These algorithms can be applied to the whole data set (for all objects in all time steps) or for subsets of the data set (e.g., for a selected set of objects, selected time periods or for selected categories). The selection is a guided process, whereat the user is provided

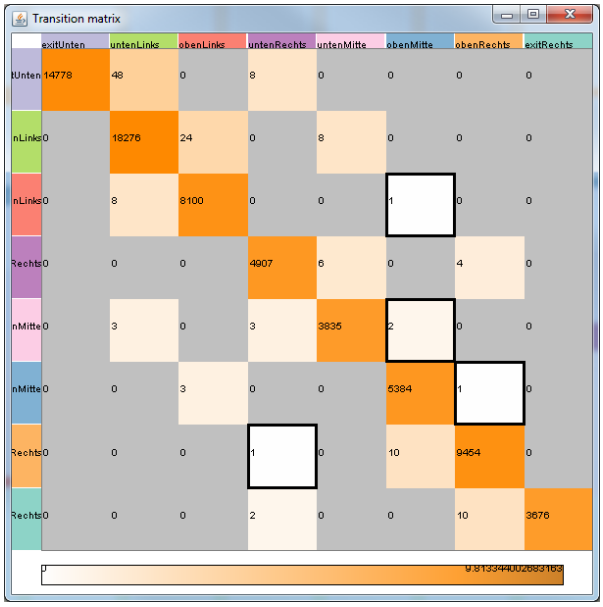


Figure 9.4.: The transition matrix is used to highlight the selected group changes. Thereby, rare transition can be identified and analyzed in more detail. In the matrix selected transitions are highlighted by a block border.

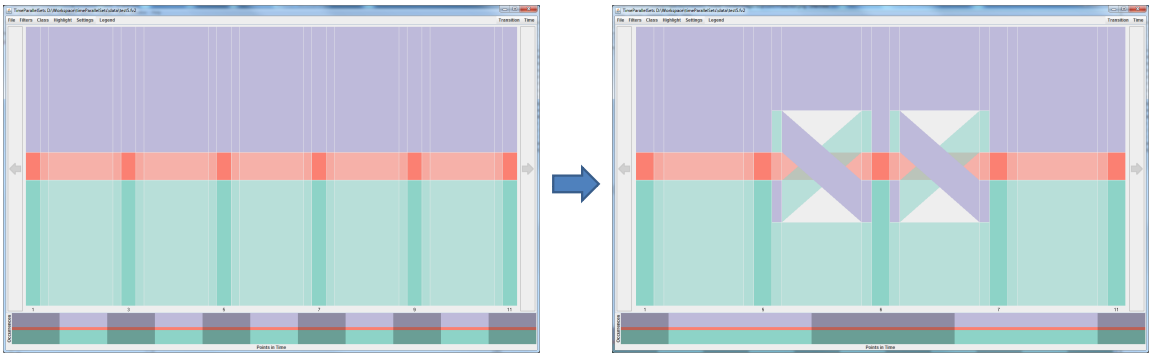


Figure 9.5.: This figure shows the importance of the selection of interesting points in time. On the left, the points in time are selected regularly. On the right, interesting points in time are detected automatically, revealing previously hidden group changes.

with visual feedback on the sensitivity of the results to the algorithm parameters. Note that the results can be adjusted interactively, where the user can additionally include in or exclude moments from the proposal.

Hence, the final choice of reference time points relies on a combination of automatic data analysis and user expertise. Algorithms for the selection of reference times require setting of parameters. A good setting of the parameters is essential, but not trivial. Small changes in parameters may have large effects on the results or vice-versa, large changes in parameters may not effect the result at all. Therefore, I follow a visual analytics approach combining automatic data analysis with interactive visualization. In particular, an interactive visualization of the automatic data analysis results for various parameters is provided. It shows the impact of parameter choice on the result. The user may then make an informed decision concerning the choice of the parameters.

In the following, new algorithms and the visual-interactive interface for time selection are presented.

9.2.2.1. Transition Weights

For task-specific time selection, the user may employ transition weights w that influence the selection of time steps. Larger weights are assigned to more important transitions.

The weights often represent distances (dissimilarities) between categories. For the presented use case (see Section 9.4), these distances can be determined by

1. geographic location of regions $L(c)$ (e.g., activity locations),
2. data values (e.g., cluster distances as Euclidean distance of cluster centers)
3. context-dependent differences between categories (e.g., shopping and work are more distant than shopping and sport).

Setting some weights to zero serves as filtering, so that irrelevant transitions are not taken into consideration.

The choice of the right weight calculation method is left to user expertise. Especially, if clustering is used, the choice of the appropriate clustering method and its parameters are not tackled in this work as I concentrate on time selection.

9.2.3. Definitions

It is assumed that the type of data containing objects has certain categorical attribute values and certain locations in different time steps. Objects are denoted as $o_i \in O$, $i = 1,\ldots,n$, where n is the number of objects in the data set. The time steps $t \in T$, $t = 1,\ldots,\tau$, are assumed to be equally spaced, or their spacing is irrelevant to the analysis. It is allowed that an object o_i in time step t_j may not exist, or the data about this object may be missing. The number of available objects at time step t_j is denoted as n_{t_j}. The number of objects existing in at least one of the time steps t_i and t_j is denoted $n_{t_i \vee t_j}$. Note that, in the data analysis, the user may focus only on a subset of time steps T^S, $T^S \subset T$ or on a subset of objects O^S, $O^S \subset O$.

It is defined that object o_i in time step t_j has a categorical value c_k (i.e., is a member of group c_k, is classified as c_k or has class c_k) as $C(o_i,t_j) = c_k$, $c_k \in C$, where $C = \bigcup_{k=1}^{m} c_k$ is the set of all categories. m is the number of distinct categories (i.e., the number of categorical values/states or classes). Moreover, it is defined a categorical value c_0 (missing), where missing or non-existing objects in specific time steps belong to $C(o_i,t_j) = c_0$. In the algorithms, the category c_0 is also included in the calculations. Therefore, C is extended as $C^0 = C \cup \{c_0\}$. For simplicity, we refer to C^0 as C, i.e., $C^0 \equiv C$.

Object o_i is defined as member of a categorical change (i.e., transition) $Tr_{c_k,c_l}^{t_1,t_2}$, when $C(o_i,t_1) = c_k \wedge C(o_i,t_2) = c_l$. This object transition is denoted as $Tr_{c_k,c_l}^{t_1,t_2}(o_i)$. Please note that t_1 and t_2 are not necessarily consecutive time steps, i.e., there may be any number of other time steps between them.

The number of objects participating in a certain transition $Tr_{c_k,c_l}^{t_1,t_2}$ is defined as $N(Tr_{c_k,c_l}^{t_1,t_2})$. The number of all objects that change categories between time steps t_1 and t_2 are defined in Equation 9.1.

$$N(Tr^{t_1,t_2}) = \sum_{c_k \in C, c_l \in c, c_k \neq c_l} (N(Tr_{c_k,c_l}^{t_1,t_2})) \tag{9.1}$$

The weight for the transition Tr_{c_k,c_l} is denoted as w_{c_k,c_l}. It is assumed that all weights are non-negative $w_{c_k,c_l} \geq 0$, and
$\exists k,l : w_{c_k,c_l} > 0$ and $mw = \max_{\forall c_k \neq c_l}(w_{c_k,c_l})$.

The weighted number of transitions is calculated as in Eq. 9.2.

$$WN(Tr^{t_1,t_2}) = \sum_{c_k \in C, c_l \in C, c_k \neq c_l} (w_{c_k,c_l} \cdot N(Tr_{c_k,c_l}^{t_1,t_2})) \tag{9.2}$$

9.2.4. Algorithms for Time Selection

The algorithms select combinations of time steps that adequately represent the data developments for the task at hand. In this respect, two types of the analytical intent of the user are distinguished:

1. Get an overview of the mainstream data developments and overall trends. A selection of time steps supporting this goal is called *globally representative*. Globally representative selections can be made according to following criteria:
   a) *Activity-based*: reveal the time periods of high and low transition activity.
   b) *Similarity-based*: reveal similarities between time steps and repetitive patterns.
2. Get information on specific facets of the dataset and reveal uncommon development patterns. A selection of time steps supporting this goal is called *focally representative*.

1. Activity-based globally representative time selection: To provide a global overview of the data, the algorithm aims at giving low prominence to low levels of transition activity and high prominence to high level of activity. The algorithm selects a subset of time steps $T^S \subset T$ so that the relative number of transitions between any two consecutive time steps within this subset $t_i^S \in T^S$ and $t_j^S \in T^S$ is approximately equal. The number of transitions is normalized, which accounts for the changes in the object sets between time steps. The same number of transitions in a small data set (small number of objects N) may be more relevant than in a large data set.

I introduce two variants of the algorithm that are based on the number of objects changing groups: one taking into account the total number of transitions between two time steps and the other looking at category-specific transitions (see Algorithm 9.2.4.1). Variant (a) relies on the total number of transitions between selected time steps. It it suitable for a general overview of the transition activity, however, it does not distinguish whether these transitions happened among few categories or were more widely distributed across categories. The category-specific number of transitions is captured in variant (b), which supposes that transitions between a pair of time steps $\{t_i,t_j\}$ are relevant if there is a significant difference between the number of transitions for all pairs of different categories (WDN). The calculation is presented in Equation 9.3.

$$WDN(Tr^{t_1,t_2,t_3,t_4}) = \frac{\sum_{c_k,c_l \in C} abs(WN(Tr^{t_1,t_2}_{c_k,c_l}) / - WN(Tr^{t_3,t_4}_{c_k,c_l}))}{(mw \cdot n_{t_1 \vee t_2 \vee t_3 \vee t_4})} \tag{9.3}$$

Note that the second variant can be used also for further selection of relevant time steps from a subset of consecutive time steps selected by the first variant (see Section 9.2.5).

Algorithm 9.2.4.1 ACTIVITY-BASED GLOBAL TIME SELECTION

$T^S = \{t_1\}$
$t_{ref} = t_1$
for all $t_i \in T, i > 1$ **do**
  a) condition on total count
  if $WN(Tr^{t_{ref},t_i}) / (mw \cdot n_{t_{ref} \vee t_i}) \geq d$ **then**
  b) condition on category-based count
  if $WDN(Tr^{t_{ref},t_{ref+1},t_i,t_{i+1}}) \geq d$ **and** $i < \tau$ **then**
    $T^S \leftarrow T^S \cup \{t_i\}$
    $t_{ref} \leftarrow t_i$
  end if{ for both conditions}
end for
$T^S \leftarrow T^S \cup \{t_\tau\}$
return T^S

The results of algorithm 9.2.4.1 depend on the setting of the parameter d. The user is provided with information on the time selection given a threshold d for an informed decision (see Section 9.2.5).

2. Similarity-based Global Selection of Time Steps This algorithm is meant to reveal repetitive patterns in object distribution across the categories, particularly, in periodic data. Data with periodic variation require specific ways of selecting representative time steps. When analyzing data for multiple periods (e.g. weather over many years), the analyst is not so much interested in changes between consecutive time steps and in differences between time steps within one period as in similarities and differences between the periods and long-time trends over many periods. For dealing with periodic data and for revealing repetitions in arbitrary data, I propose similarity-based time selection, which employs clustering of time steps by similarity of the corresponding data. The result is a set of time clusters, each containing a subset of time steps $T^K = \{T^{K_1}, {}^{K_k}\}, T^{K_i} = \bigcup t_j^{k_i}$. Each time cluster gives a selection of time steps to be interactively explored. The user needs to explore all time clusters to construct a full picture of the data developments.
For the time step clustering, the distance (amount of dissimilarity) between time steps can be measured as dissimilarity of the object classes taking into account their weights (see 9.2.2.1). The choice of clustering method and its parameters is user-defined. Several clustering algorithms (e.g., K-Means , SOM, DBScan) [WFH11] are provided for this purpose.

3. Activity-based Focal Selection of Time Steps: This algorithm is used when the user wants to focus on particular data developments. There are two variants, which search for pairs of time steps with either very high or very low transition activity (the latter case means irregular transitions, i.e., outliers). This algorithm selects a set of time pairs formed by consecutive time steps $T^S = \bigcup\{t_i, t_{i+1}\}$, where the weighted number of transitions is higher/lower then a threshold d (see Algorithm 9.2.4.3a and b). The threshold d is user-defined using a visual-interactive interface (see Section 9.2.5). Please note that in case of searching for outliers in transitions, the

Algorithm 9.2.4.2 SIMILARITY-BASED GLOBAL TIME SELECTION

```
Determine distances of time steps
for all t_i ∈ T do
  for all t_j ∈ T, i ≤ j do
    dist(t_i,t_j) = Σ_{c_k∈C,c_l∈C}(w_{c_k,c_l} · N(Tr^{t_i,t_j}_{c_k,c_l}))
  end for
end for
T^K ← cluster(T)
return T^K
```

normalization of transition count is not used (e.g., when only one object changes categories c_k, c_l). Weights for finding unusual transitions are not used, apart from filtering out irrelevant transitions (i.e., with zero weight).

Algorithm 9.2.4.3 ACTIVITY-BASED FOCAL TIME SELECTION

```
T^S = ∅
for all t_i ∈ T, i < τ do
  a) condition for high activity
  if WN(Tr^{t_i,t_{i+1}})/(mw · n_{t_i∨t_{i+1}}) ≥ d then
  b) condition for low activity
  if 0 < N(Tr^{t_i,t_{i+1}}_{c_k,c_l}) ≤ d and w_{c_k,c_l} > 0 then
    T^S ← T^S ∪ {t_i, t_{i+1}}
  end if{for both conditions}
end for
return T^S
```

9.2.5. Visual-Interactive Threshold Selection

The setting of proper parameters for algorithmic data analysis is usually essential for gaining high quality results. A good practice is to inform the user about the effect of the possible parameter choices on the result. In TIPS, the selection of time steps in Algorithms 9.2.4.1 and 9.2.4.3 depends on the parameter d. If d is too small, all time steps can be selected. If d is too large, none or only the first and last time steps are selected. The user is provided with information on the distribution of the selected time steps depending on threshold d for an informed decision.

For the threshold selection view, a set of threshold values $d \in D$ is determined, for which the selected time sets $T^S(d)$ are calculated. Thresholds $d = d_0 + i * d_s$ are used, $i = 0, \ldots, i_{max}$, where d_0 is an initial threshold, d_s is the step between threshold values and i_{max} is the stopping criterion. The values of d_0 and d_s depend on the applied algorithm. For the algorithms 9.2.4.1 and 9.2.4.3, where $0 \leq d \leq 1$, small values of the initial threshold and the step: $d_0 = 0.05$ and $d_s = 0.05$ are choosen. The number of time steps in algorithms9.2.4.1 and 9.2.4.3 is generally non-increasing with growing d. Note that the Algorithm 9.2.4.1 a) selects time steps in a successive manner with a fixed addition of the last step. There can be special cases where a time step is added close to the end leading to an increase in the selected time steps by one when using larger d.

In the algorithm 9.2.4.1b), the number of selected time steps decreases with increasing d, as atypical events in the data ($\ldots \leq d$) are from interest. The calculation starts with a high d_0 and decreases it constantly. $d_0 = (int)0.05 * n$ and $d_s = -1$.The choice of d_0 is motivated by the general rule of thumb that $\leq 5\%$ threshold is

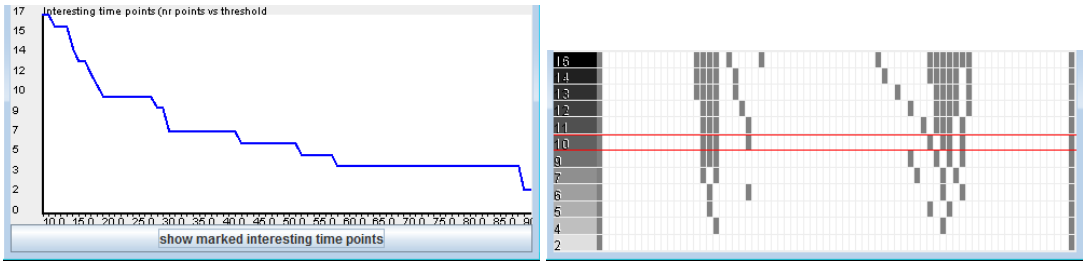


Figure 9.6.: Choosing a threshold for time selection algorithms. Left: The impact of the threshold on the number of selected time steps. X-axis: threshold value, y-axis: number of selected time steps. Right: The impact of the threshold on the temporal distribution of the selected time steps. X-axis: All time steps, where selected time steps are shown in grey. Y-Axis: number of selected time steps. The setting highlighted in red is chosen for visualization.

commonly assumed as atypical [KKZ09]. The i_{max} determining the stopping of i increase is determined by the data – when there are no relevant times selected, because the threshold is too high (too low in case of Algo. 3*b*).

The user is provided with a view on the relationship between the threshold and the number of time steps selected by the algorithm (see Figure 9.6 top). This view was inspired by the work on selection of dimension reduction threshold [JJ09]. The parameter setting is difficult, as a small change of the threshold may result in a large decline of the number of time steps and vice versa. By looking at the line chart, the user may choose a relevant number of times by, e.g., "elbow criterion" [KS96], which is often used for determining the number of clusters as the point where adding more clusters does not add much information. This view, however, does not show a deeper insight into the distribution of time steps across the data set. This is tackled in a new view described below.

The new visualization shows *details on the impact of threshold setting on the time selection* (see Figure 9.6 right). It provides information both on the number of selected time steps for threshold d_i and on their positions. The number of selected time steps is shown in a gray color scale on the left together with the value of the threshold. The selected time steps are shown on the right as gray-shaded rectangles. Please note that only those thresholds are shown, that give different results in comparison to the previous threshold values. The motivation for this design choice can be seen in Figure 9.6 left. Many changes of threshold do not lead to changes in time selection. Therefore, to compress this view, only different time selections are displayed. Moreover, this filtering also compensates for non-optimal selections of the initial threshold d_0 and the step d_s.

Calculation complexity in the algorithms 9.2.4.1 and 9.2.4.3 is determined by the number of objects, number of time steps and thresholds used for final calculation ($O(N \cdot T \cdot D)$, where D is the number of thresholds used for selection proposals). Note that the assessment of the calculation complexity for calculating clusters in algorithm 9.2.4.2 goes beyond scope of the paper. For more details regarding the complexity of cluster calculation, which is the majority of the computational expense, please see [WFH11].

9.3. Application to Emergency Evacuation

The usefulness of my approach is demonstrated on the "IEEE VAST 2008 Challenge – mini challenge 4: evacuation data" [GPL*08]. The dataset describes the movement of 82 subjects in a building over 837 points in

time [AA10]. I partitioned the building into eight areas (see Figure 9.7 left) and put the subjects into groups according to their location in every moment. The group coloring can be seen in Figure 9.7 right. At a specific time, a bomb detonates and afterwards people die or start to move towards the exits (turquoise and purple areas).

The input data is visualized on the basis of 15 automatically recommended time points, which are highlighted in the overview on the bottom (see Figure 9.8(a)). The distribution of the selected moments is unequal over the whole time period. The automatic analysis using the algorithm for the activity-based global time selection (see Algorithm 9.2.4.1) puts more emphasis on time periods of high movements (after the explosion). As can be seen, most people reached one of the two exits. To reveal the escape routes, the subjects arriving in the particular exits are highlighted. People using the bottom exits at the beginning were mostly located in the upper left or one of the bottom regions. Many people as expected move towards the nearest exit, e.g., from the top left (red) to the bottom left (green) region. Surprisingly, even some people from the top middle region (blue) preferred the bottom exit to the closer right exit (see Figure 9.8(b)).

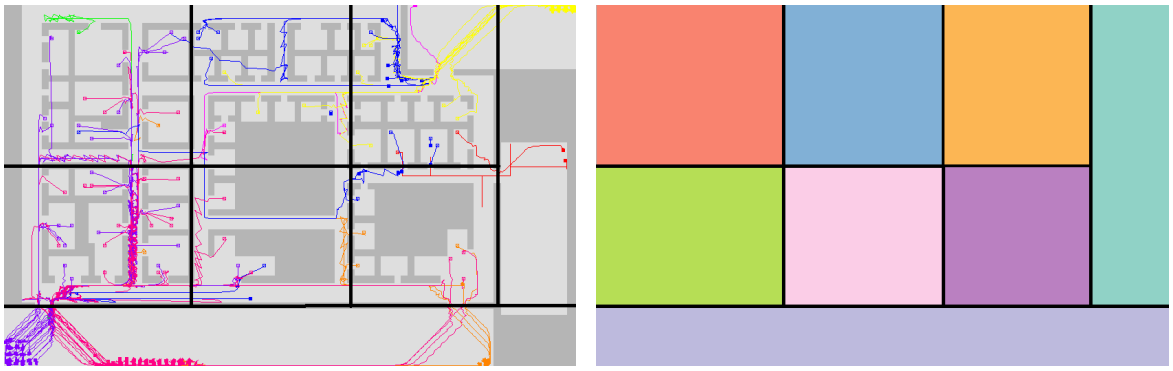
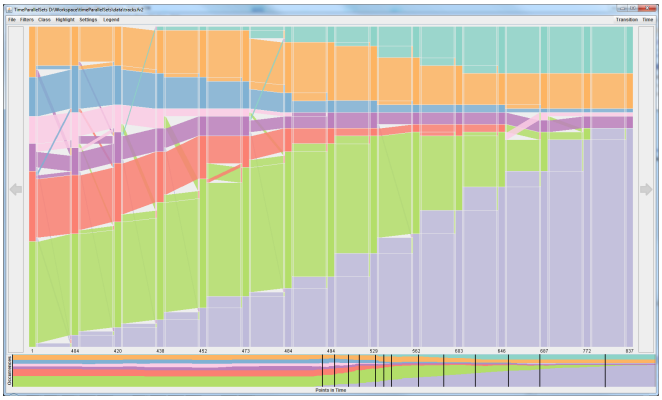


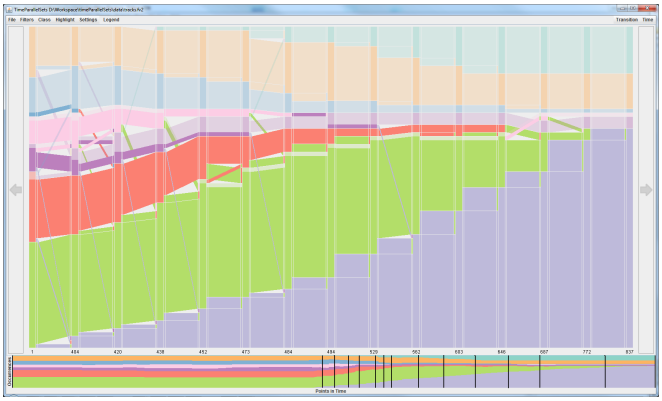
Figure 9.7.: The example dataset from the VAST challenge 2008 [GPL*08]. (a) Trajectory based visualization of the movement of the people [AA10] with partitioning of the building into 8 areas. (b) Coloring of the areas.

If a security analyst wants to improve the escape routs, she can first highlight all people, who remain in the upper right (orange) area after the evacuation (see Figure 9.8(c)). Some of them have never moved (probable victims of the bomb). Interestingly, there is one person who came from the upper mid (red) area. She did not take the shortest path so she did not manage to exit the building fast enough.

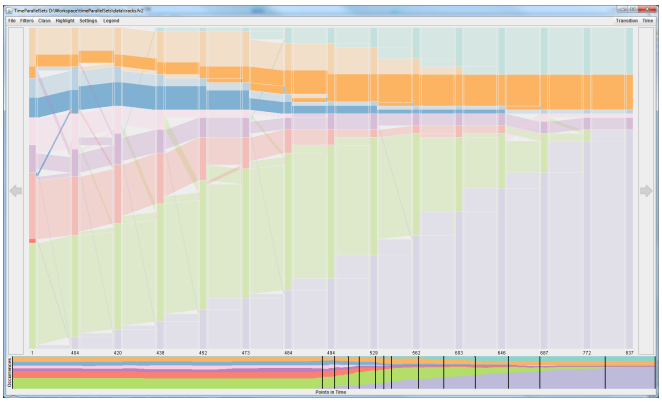
The analyst can then focus on the analysis of the behavior of people who move differently from the rest or in an unexpected way (away from exits). Those few region changes are selected in the transition matrix (see Figure 9.4) and are then highlighted in the main view (see Figure 9.9(a)). As it turns out, only some of the marked transitions (orange to blue and blue to red) are visible in the initial view (using automatic time selection). The other transitions (purple to orange and blue to pink) are not captured by the selected points in time. Therefore, the selection of the representative time points was recalculated, ensuring to show all selected transitions and at the same time achieving a overview about the whole data with the desired 15 key points in total (see Figure 9.9(b)). Tracking these people reveals that despite their odd routing, the majority reaches the exits (see Figure 9.9(c)).



(a)

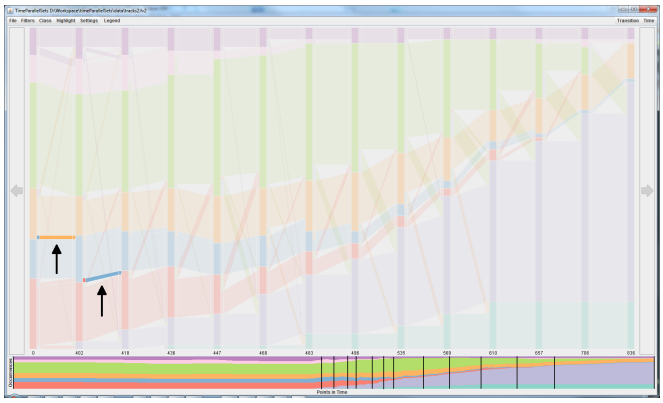


(b)

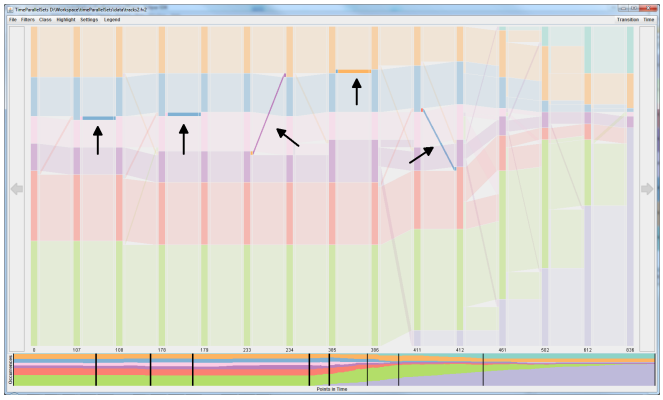


(c)

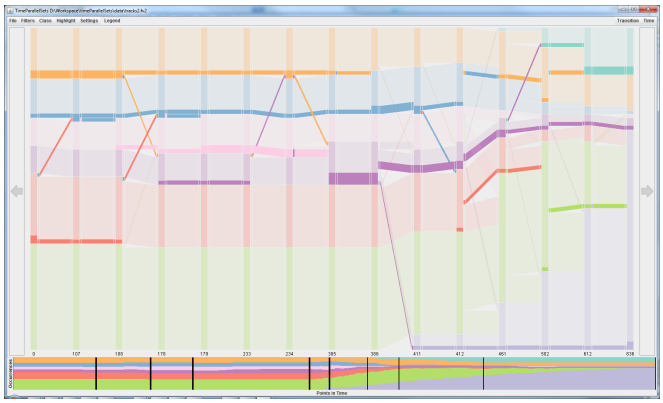
Figure 9.8.: Visualization of the evacuation data. a) Data visualization using relevance-based time selection. b) Tracking of people who reach the exit on the bottom. c) Tracking of people who were located in orange area at the end.



(a)



(b)



(c)

Figure 9.9.: Analysis of unexpected transitions. a) Initial time selection shows only some of the selected transitions. b) Overview of interesting transitions using transition-based time selection. c) Tracking of selected transitions. Despite the unexpected movement, most of the highlighted people reach the exit in time.

(a) Milan regions (b) Trajectories on Wednesday (c) Trajectories on Sunday

Figure 9.10.: Use case data. Milan has been divided into geographic regions forming categories (a). Mobile phone caller trajectories on Wednesday (b) and Sunday (c).

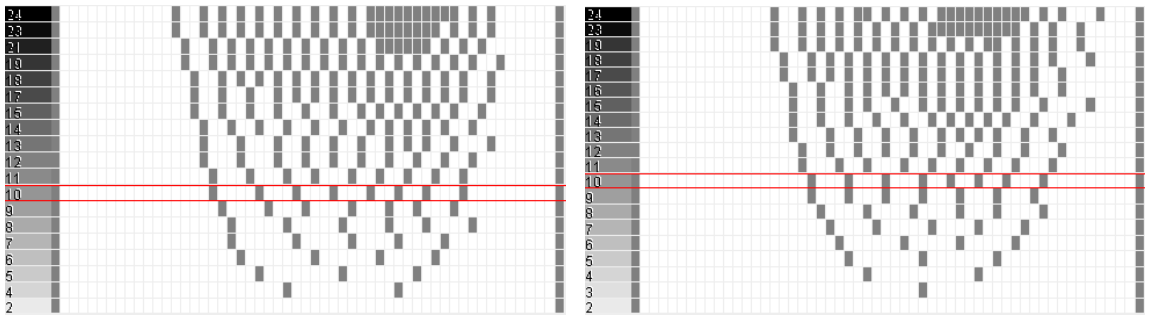
9.4. Application to People Movement

As a second example, the proposed approach is applied to mobile phone call records . This was done in cooperation with Natalia and Gennady Andrienko from Fraunhofer Institute IAIS, Bonn, who provided the data and their geo-visualization toolkit. The dataset contains 5,108,298 mobile phone call records from 367,730 customers of an Italian mobile phone company WIND collected over a period of 9 days in Milan area. From these data, we have constructed caller trajectories and divided them into daily tracks. The territory of Milan has been split into 307 Voronoi polygons built around the positions of the network antennas. Since these polygons are not meaningful in this case, we grouped them into larger regions reflecting the geography of the city (see Figure 9.10).

To explore and compare the mobility of the phone users on a typical working day and at the weekend, we select two subsets of trajectories from the database: from Wednesday and from Sunday. We also remove the trajectories of stationary customers who made all their calls in the same cell or a few neighboring cells. This gives us 6,943 trajectories of mobile customers on Wednesday and 4,496 on Sunday. It signifies lower calling activity at the weekend. From these trajectories, we generate time series of visited regions with a time step of 30 minutes and visualize them in TIPS.

With the help of tools for interactive threshold selection, we select 10 globally representative time steps for Wednesday and Sunday (see Figure 9.11). The selected time points indicate a later start of activity on Sunday than on Wednesday.

The behavior of callers between the selected time steps is shown in TIPS (see Figure 9.12). The colors correspond to the city division. Grey color means that the locations of the callers are unknown; this occurs

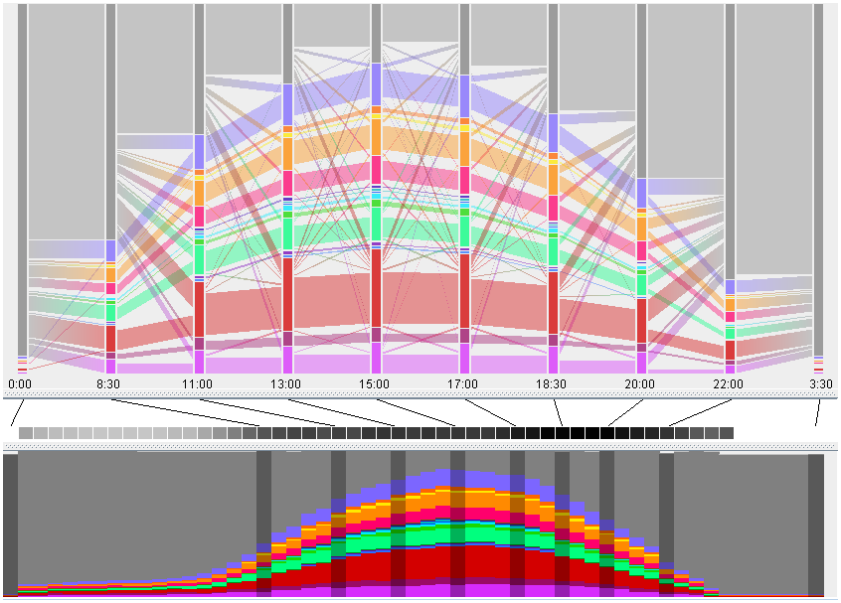


(a) Wednesday (b) Sunday

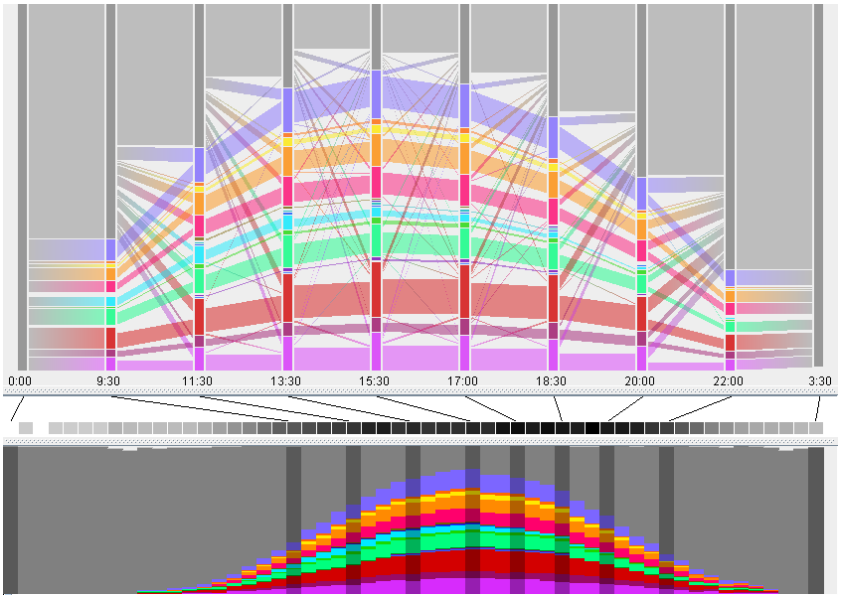
Figure 9.11.: The overviews of the selected time moments with respect to the choice of threshold. It shows that the majority of phone call activity on Sunday (b) start later and finish than on Wednesday (a).

before the first call and after the last call. Main activity starts later on Sunday (by 9:30) then on Wednesday (by 8:30), based on globally relevant time point selection. Movement pattern between areas is very similar on both days, although time spans of these activities differ. The most visible flows are from “unknown” and to “unknown”, while the flows between regions are much smaller. The summary view on the bottom shows: On Sunday compared to Wednesday, the presence of people by regions is proportionally lower in the Center (red) and Linate (green) and proportionally higher in e.g. San Siro (purplish red) and Milano 2 (cyan). The presence in Milano 2 is notably higher in the first half of the day.

A detailed inspection using TIPS (see Figure 9.12b) revealed, that there are much more activities on Sunday than on Wednesday in Milano 2 (cyan). Hereupon, we have used interactive selection of trajectories by visited areas and thus created a new instance of TIPS with the selected subset. On Wednesday, the presence of people in this region was rather constant, and there were no unusual change patterns (no focally representative time steps were detected). On Sunday, something particular was going on in Milano 2 in the first half of the day. Using the link to the map display, we select the trajectories of the people who stayed in Milano 2 at the selected representative time moments presented in Figure 9.13 top. Figure 9.13 left shows these trajectories. All or almost all trajectories meet in a single point. Hence, we can identify the place more precisely where unusual activities occurred on Sunday. Figure 9.13 right shows this place and its surrounding with more geographical detail. The place is near the metro station Cascina Gobba. By searching on the Internet for the possible reason of the unusual activities, we found several videos in YouTube showing the flea market at Cascina Gobba, which explains our finding.

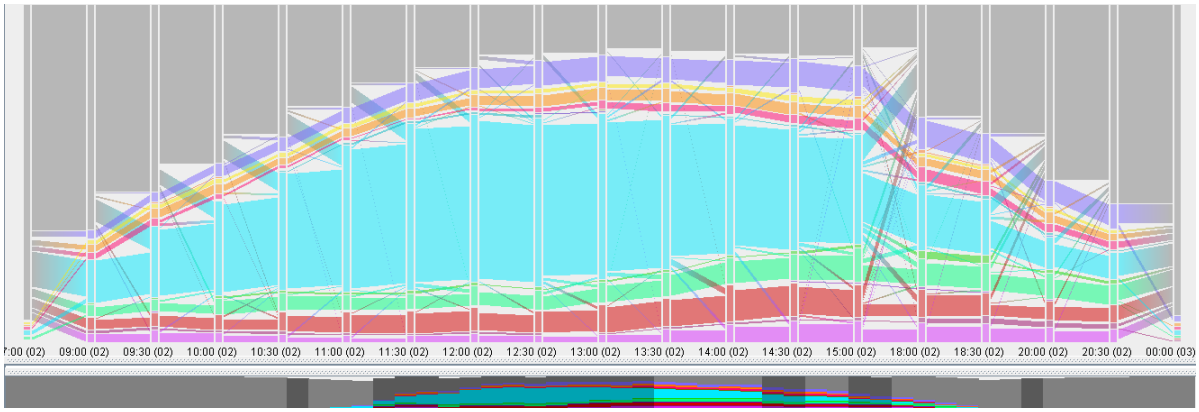


(a) Wednesday

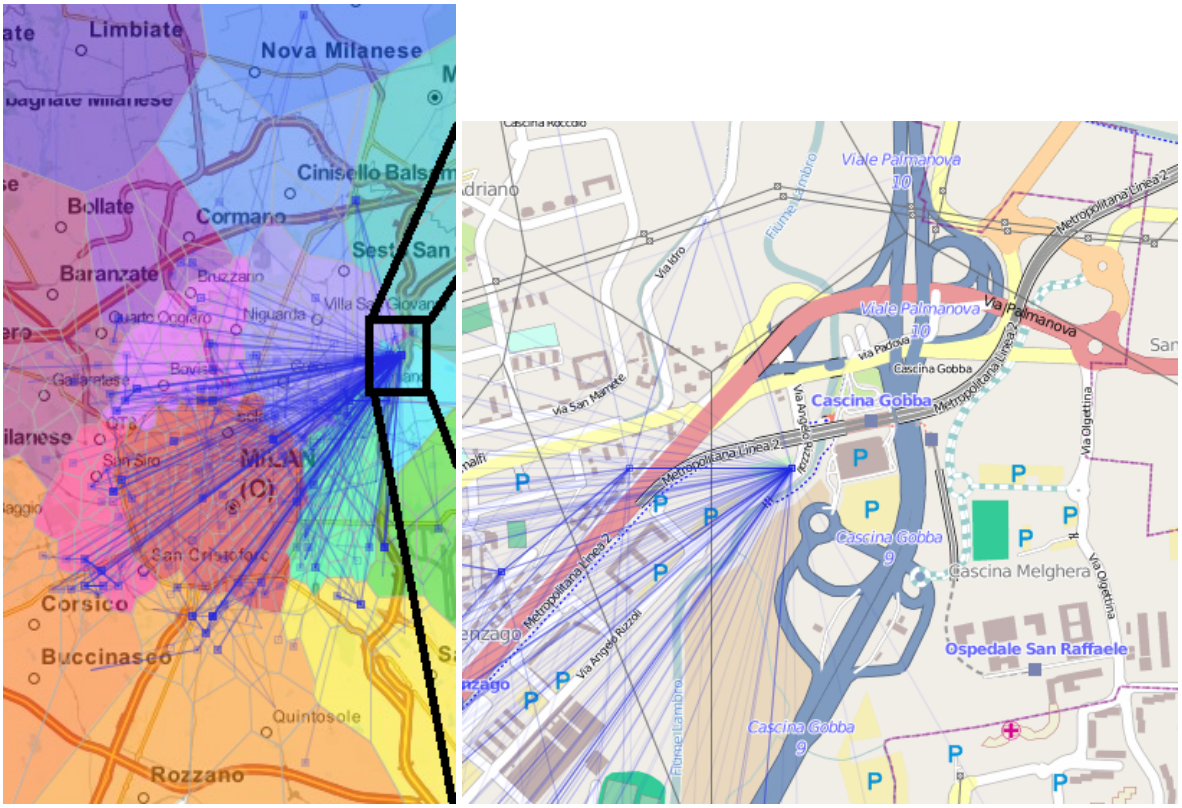


(b) Sunday

Figure 9.12.: Daily movements for Wednesday (a) and Sunday (b). The figure allows to compare the calling and movement behavior of people between a weekday and at weekend. It shows, that the majority of phone call activity on Sunday start later then on Wednesday. Although the general movement patterns show high similarities, the proportion of people in regions varies, especially in the center of Milan (red) and in Milano 2 area (cyan).



(a) Representative time moments



(b) Trajectories for Milano 2 area

Figure 9.13.: Visualization of trajectories of people visiting Milano 2 area (in cyan) on Sunday. a) TIPS view with representative time steps. b) Trajectories of the people who were in Milano 2 at these time steps. Right: Zoom into the area where all trajectories meet.

10. Explorative Sequence Analysis

In this chapter, an interactive visualization tool for the analysis of large and complex time sequences is introduced. It combines multiple linked views, each specialized on the analysis of other aspects of the time sequence. Time sequences are an symbolic representation to reduce the complexity e.g., of long or high dimensional time series. Typically, similar patterns are represented by only one symbol, again reducing complexity. Sequences of these symbols represent the source data and are expected as input data of the approach. The tool is presented using time series data as exemplary input. The preprocessing step converting the source data to sequences relies on an idea of [SBVLK09] and is briefly presented in Section 10.2.1 for a better motivation of the usecase. Subsequently, the algorithmic part covering the similarity calculation of subsequences (see Section 10.2.2) and the different visualizations (see Section 10.2.3) are described in detail. Section 10.3 describes an exemplary workflow analyzing the previously introduced financial data.

The work presented in this chapter is partially based on [BMvLS09].

10.1. Related Work

The idea of converting a continuous time series to a symbolic sequence has been proposed by Lin et al. [LKLC03]. They split the continuous range of an one dimensional time series in a few blocks with a previously defined duration. Thereby, the data can be compressed dramatically, which allows the fast and efficient handling of large databases [SK08]. They extended their approach with a tree like visualization where each subsequence is represented by a path from the root to the leaves. This visualization scales well with the number of subsequences but not with their length or the number of available symbols [LKL05]. My work is based on a SOM-based framework to discretize high dimensional time series which was proposed by Schreck et al. [SBVLK09]. For more details please see Section 10.2. Recently, Ward and Guo presented an approach, which projects high dimensional time series into the two dimensional space using multidimensional scaling [WG11]. Similar moments in time are placed nearby, which are connected according to the time flow. Thereby, similar subsequences form similar paths and enable and intuitive comparison of sequences. However, the comparison of many sequences might lead to overplotting.

A lot of work for analyzing sequences has been done in the biological area. They typically do not focus on time dependent, but on DNA or protein sequences e.g. BLAST [AGM*90] for the comparison of two or Clustal [CSK*03] for the comparison of multiple sequences. In the analysis of biological sequences, it is convenient to insert gaps for a better alignment of local patterns. This does not work for time dependent data as the temporal component is an important and immutable component of the analysis.

In the text mining domain, a lot of work has been done on the analysis of sequential patterns. For example, Wong et al. [WCF*00] introduced an efficient algorithm to search for frequent subsequences in large texts which we use in our approach, too. However, the time component is missing in the text mining cases as well.

10.2. Approach

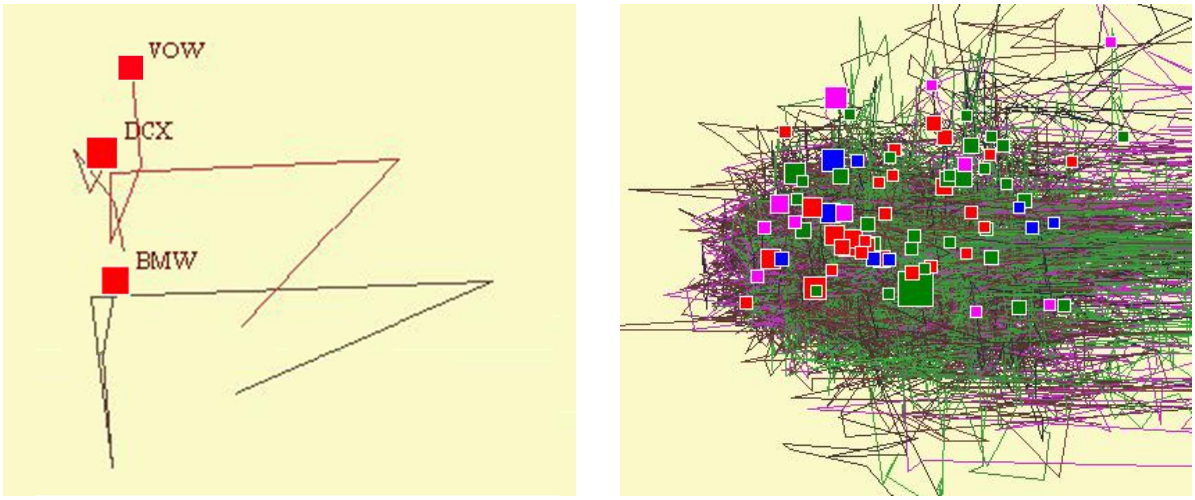
The approach combines multiple visualizations to explore the similarities and differences of the input data. After the preprocessing and an user steerable pairwise similarity definition of the symbols, the user can analyze the data in different visualizations, each specialized on other questionings. The strength of the respective parts are combined by linking all visualizations to form a bigger picture of the analyzed data set.

10.2.1. Input Data and Preprocessing

This approach is designed for the general analysis of sequenced time depended data. For the illustration of the approach, example data from the financial sector is used. The generation and preprocessing of this data is described by Schreck et al. [STKF07, SBVLK09] and is not part of this thesis. However, their approach is described brifely for a better understanding of the used data. The datasets contains risk estimations and return values for the stocks of 60 companies over a period of 500 days. The goal of the explorative analysis is to find patterns of interest. The characteristics and/or the particular interestingness of these patterns is not known á priori as it heavily depends on the analyzed usecase and dataset. Examples for such patterns in the given scenario could be either the similar behavior of multiple companies within a given time period or single companies developing against the trend or to identify periodicity of discovered patterns. A common approach for the analysis of two dimensional data is using a scatter plot. In case of two dimensional time dependent data, the time aspect can be visualized using animation or, to allow an better overview, trajectories. This works well for small datasets with only few objects and short time periods (see Figure 10.1(a)) In large datasets, the display gets crowded quickly (see Figure 10.1(b)). To reduce the complexity of the data, the time series were spitted into smaller chunks of five days to be in accordance with the length of one trading week. A few weeks with missing values e.g. caused by holidays were discarded. Afterwards, each five-day-trajectory was normalized to have the same minimal and maximum values. The reasoning behind the normalization is based on the particular question the analyst wants to answer. External factors (e.g. increasing costs of a resource) have a smaller influence on big companies with a high market capitalization than on small companies. Hence, without normalization, similar reactions e.g. in the risk assessment would not be comparable. To decrease the complexity furthermore, representative patterns were calculated using a SOM projection [Koh01]. Each neuron is represented by a 10-dimensional vector according to the input trajectories (two dimensional position on each of the five points in time). After the training process, each neuron represents several input trajectories and can be directly visualized as trajectory. As the neurons are organized in a two dimensional network, the result of the training is a two dimensional map visualizing the prototypes which are sorted according to their local similarity (see Figure 10.2). For these SOM calculations, the framework of Schreck et al. [SBVLK09] was used. Besides the interactive visualization and parameter steering, it provides several quality measures to evaluate the projection quality.

10.2.2. Definition of Similarity

The input data contains several time depended sequences of symbols. In the exemplary usecase, each sequence is represented with the reduced set of different trajectories, based on the prototypes of the SOM calculation. The goal of the analysis is always the identification of patterns in sequences of symbols. Each pattern can be represented by an subsequence of arbitrary length between 1 and the maximum length of a sequence. For the comparison of two (sub)sequences, an appropriate distance measure has to be defined. On the lowest level, this is the distance between two symbols. In the financial case, it is represented by the euclidean distance between the 10 dimensional vectors of the prototypes. Other scenarios e.g. the comparison of amino acids in protein sequences use non symmetric distances according to mutation probabilities acquired in biological experiments [HH92]. To



(a) (b)

Figure 10.1.: a) Trajectories can be used to represent the movement of the objects over time. In this usecase, each company is visualized according to risk estimations and return values of their stocks. b) The visualization of many objects over longer time periods leads to an massive overplotting effect.

give the users the possibility to integrate their domain knowledge in the analysis process, they can interactively change and define the automatically preprocessed distances. Using the SOM visualization, similarities of the symbols can be redefined (see Figure 10.3). Distances can be set for each pair of symbols separately or using predefined masks like radial kernels. A for one object defined distance mask can then be generalized to all other objects. Additionally, groups of similar objects can be defined, reducing the set of symbols used in the following calculations and visualizations.

The distance of two subsequences is defined as the sum of the pairwise distances of the symbols at the particular position. To increase scalability, the number of occurrence of a substructure within the dataset has to pass a user defined threshold. The used algorithm is based on an idea by Wong et al. [WCF*00], who use it for text analysis. The result is a lexicographic tree containing all subsequences passing the choosen threshold.

10.2.3. Interactive Sequence Analysis

The sequence analysis tool combines different interactive visualizations, each representing other aspects of the data.

- The **sequence view** offers an overview of the sequences and supports their sorting according to several aspects.
- The **heatmaps** allow a compressed view on the frequency and distribution of patterns over time or over the sequences.
- The **scatter plot** visualizes the relationship of different, user selected attributes.

In all views, the visualized subsequences can be filtered according to length and number of occurrence. In the following, these views and their connections are described in detail.

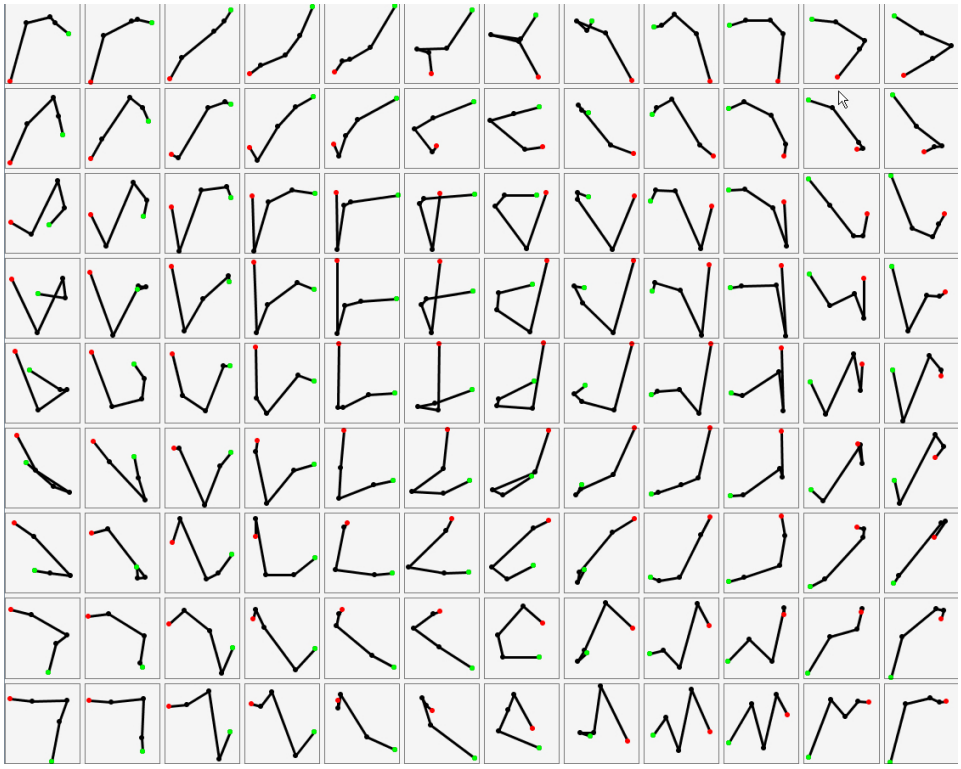
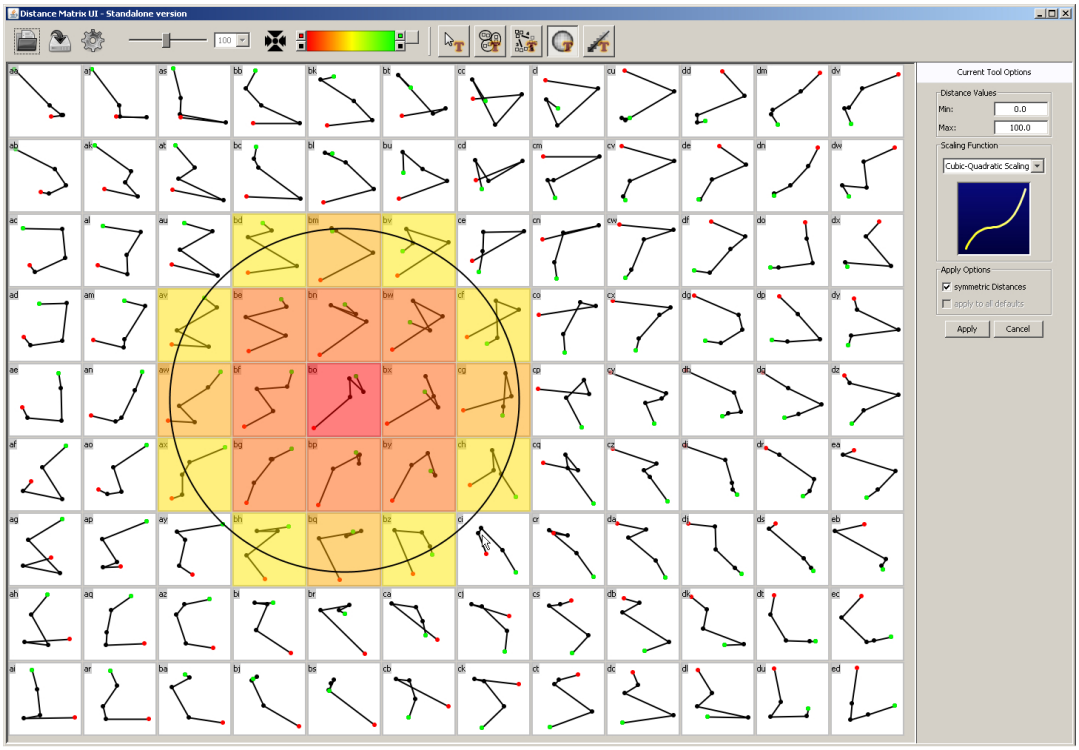
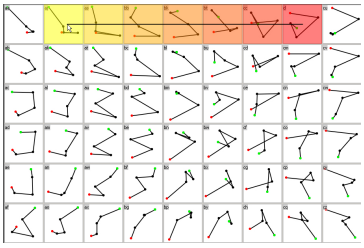


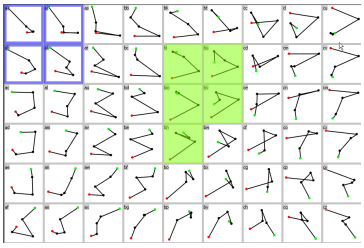
Figure 10.2.: This figure shows the result of the SOM calculation with the weekly stock patterns as input. The green dot represents the first day, the red dot the fifth day. Each cell shows a prototype which represents several input objects, whereas the cells are sorted according to local similarity.



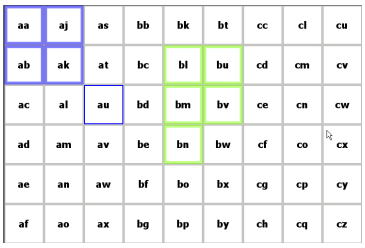
(a)



(b)



(c)



(d)

Figure 10.3.: This figure shows the visualization for the interactive definition of distances between symbols. It is based on the results of the SOM calculations. The distances can not only be defined pairwise but different masks which represent similarity patterns can be used. a) Circular mask with cubic-quadratic scaling b) Line mask with linear scaling c) Grouping of objects d) Abstract visualization with alphabetical representation of each object.

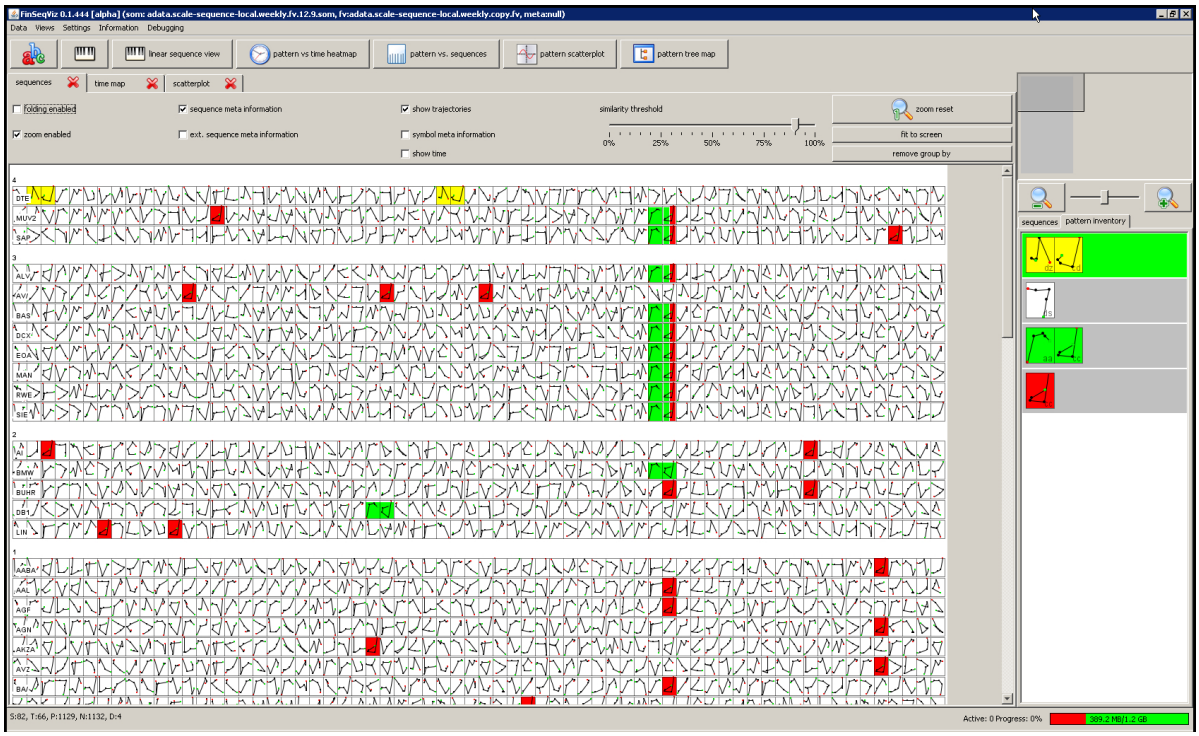


Figure 10.4.: The sequence view visualizes the complete input sequences. Selected subsequences can be highlighted. All in this or in other views selected subsequences are listed on the right side of the display.

10.2.3.1. Sequence View

This view visualizes the input sequences (see Figure 10.4). The symbols can either be shown as abstract letter combination or using an arbitrary renderer to visualize e.g. trajectories. As default, the sequence of each object is shown in one line and each column represents one point in time. To get more detailed information, zooming with and without line wrapping is included. The objects can be sorted by occurrence of selected subsequences or according to arbitrary meta data e.g. in the case of financial data, key data like market capitalization or number of employees. Subsequences which can be selected in this and the other views are listed on the right side for further investigation.

10.2.3.2. Heatmap View

The heapmap view offers a compact visualization of the dataset and can be used to identify e.g. outliers (see Figure 10.5). One axis represents all patterns of the dataset, the other represents all time points or sequences. The bars represent the frequency of the respective pattern respectively the number of difference patterns at a given time point. Each cell in this matrix represents a subsequence at the respective point in time, whereas the number of occurrence is mapped to color ranging from blue (zero) to yellow (max). The patterns are then sorted according to their number of occurrence at the a selected point in time (see Fig 10.5(b) and 10.5(c)).

Alternatively the time can be replaced by the sequence id to analyze the frequency of patterns over all sequences (see Fig 10.5(b)).

10.2.3.3. Scatterplot View

The scatterplot offers an easy and intuitive way to evaluate the dependency of the patterns of selected variables. Selected points are indicated by uniquely colored borders. The three displayed variables (mapped to x-axis, y-axis and color) on the axis can be divided into six categories and individually selected by the user.

- *Data specific* properties which depend on the used dataset e.g. in this usecase maximal risk or average return value. If analyzing biological sequences, this could be chemical properties like charge or solubility.
- *Frequency of the subsequences*
- *Length of the subsequences*
- *Internal similarity* reflecting the similarity of the respective symbols within each subsequence. This value is only displayed for subsequence with a minimal length of two as it is calculated as the average of the pairwise distances of all contained symbols.
- *Time related* characteristics e.g. the variance or absolute frequency of the points in time where the respective subsequence occurs.
- *Sequence related* properties like the number of sequences containing the respective pattern or the average occurrence per sequence.

10.3. Application to Financial Data

In the following, an exemplary workflow based on the previously introduced financial dataset is presented. Each trajectory symbol represents the progression of risk assessment and return evaluation over five days. The total length of the analyzed time series covers 500 days resulting in 100 5-day-trajectories. As starting point for the analysis, getting an overview is advantageous. First, a view on the heatmap reveals a quite prominent outlier shown by a bright yellow pixel (see Figure 10.5(a)). It is selected as interesting and stored in the pattern list on the right for further analysis e.g. to get more details about the distribution in the sequence view. Next, the scatter plot gets attention, selecting frequency on the y-axis and the average frequency per sequence respectively per company on the x-axis. The color represent the particular length of the subsequences. The two axes exhibit a strong correlation, so short subsequences appear more often per company than long ones, which was to be expected (see Figure 10.7). Again, outliers are selected: The subsequence which occurs most often (blue), which has the highest frequency per sequence (yellow) and one of the most unique subsequences (green). The previously in the heatmap selected pattern was automatically highlighted in red. With this list of potentially interesting patterns, the analyst switches to the sequence view. First, the blue, most often occurring symbol is highlighted but it appears to be randomly distributed (see Figure 10.8(a)). Next, the other three patterns are highlighted and the sequences are sorted according the frequency of their occurrence. A strong distribution pattern is visible (see Figure 10.8(b)). The red trajectory is part of the green one and they show a increased appearance at the 46 & 47 week. Thus, many companies exhibit similar characteristics at the same point in time, which can serve as a starting point for a deeper financial analysis. The yellow pattern is only visible in one company, but there it appears twice. Again, this unique characteristics of the company is from interest for the financial analyst. With this sequence analysis tool, the users can analyze large, time dependent datasets. The combination of the different strength of the linked views allow a identification and drill down to potentially interesting patterns.

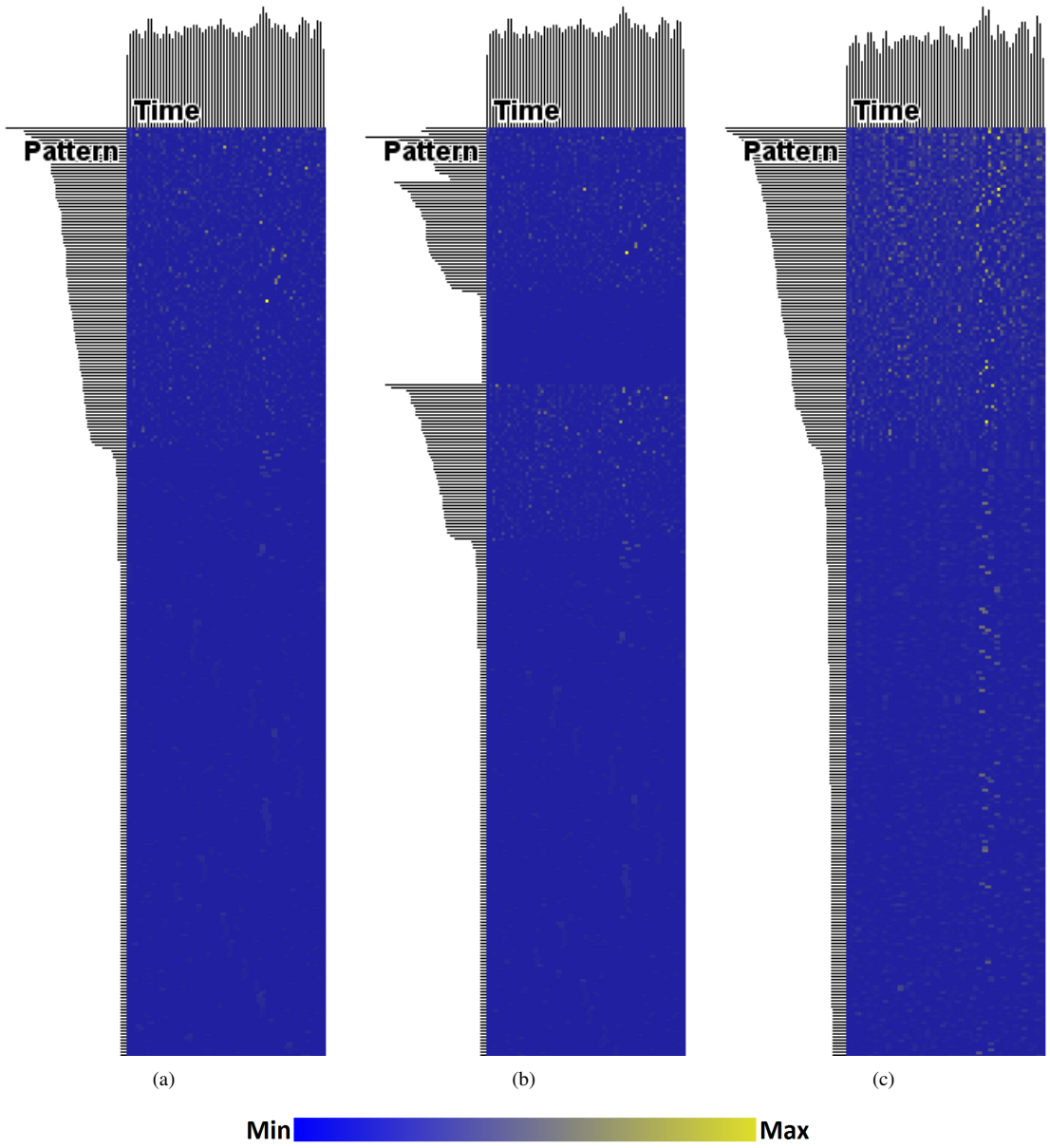


Figure 10.5.: Heatmap View: Each heatmap visualizes the number of occurrence of each subsequence at each time point and ranges from zero (blue) to the respective maximum (yellow). The bars represent the accumulated frequency of the respective column or line. a) Ordering according to the total frequency of a pattern. b) Ordering according to the frequency in the selected point in time t1 c) Including patters within a selected similarity threshold.

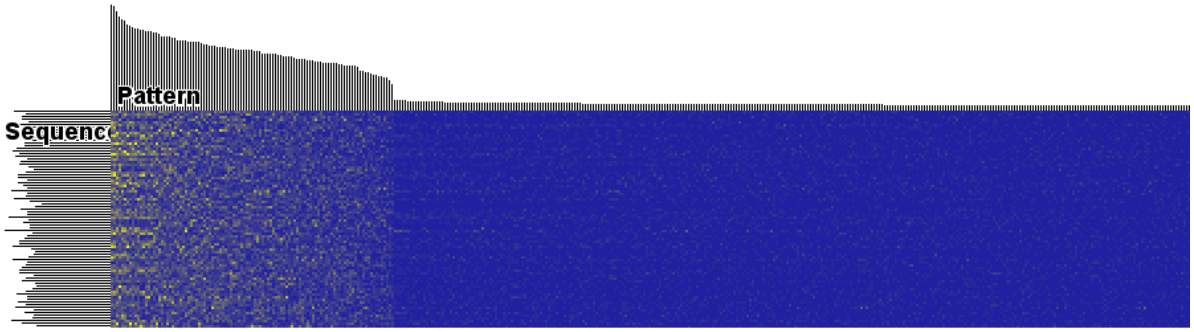


Figure 10.6.: Heatmap View: The heatmap visualizes the number of occurrence of each subsequence within each complete input sequence.

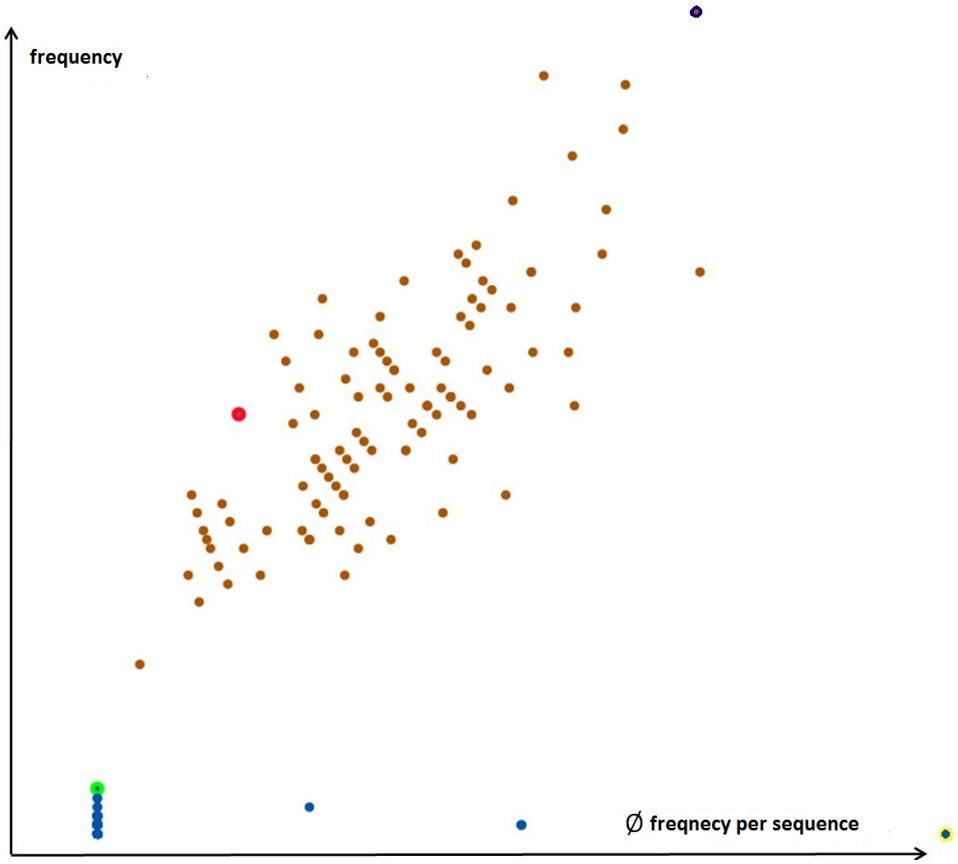
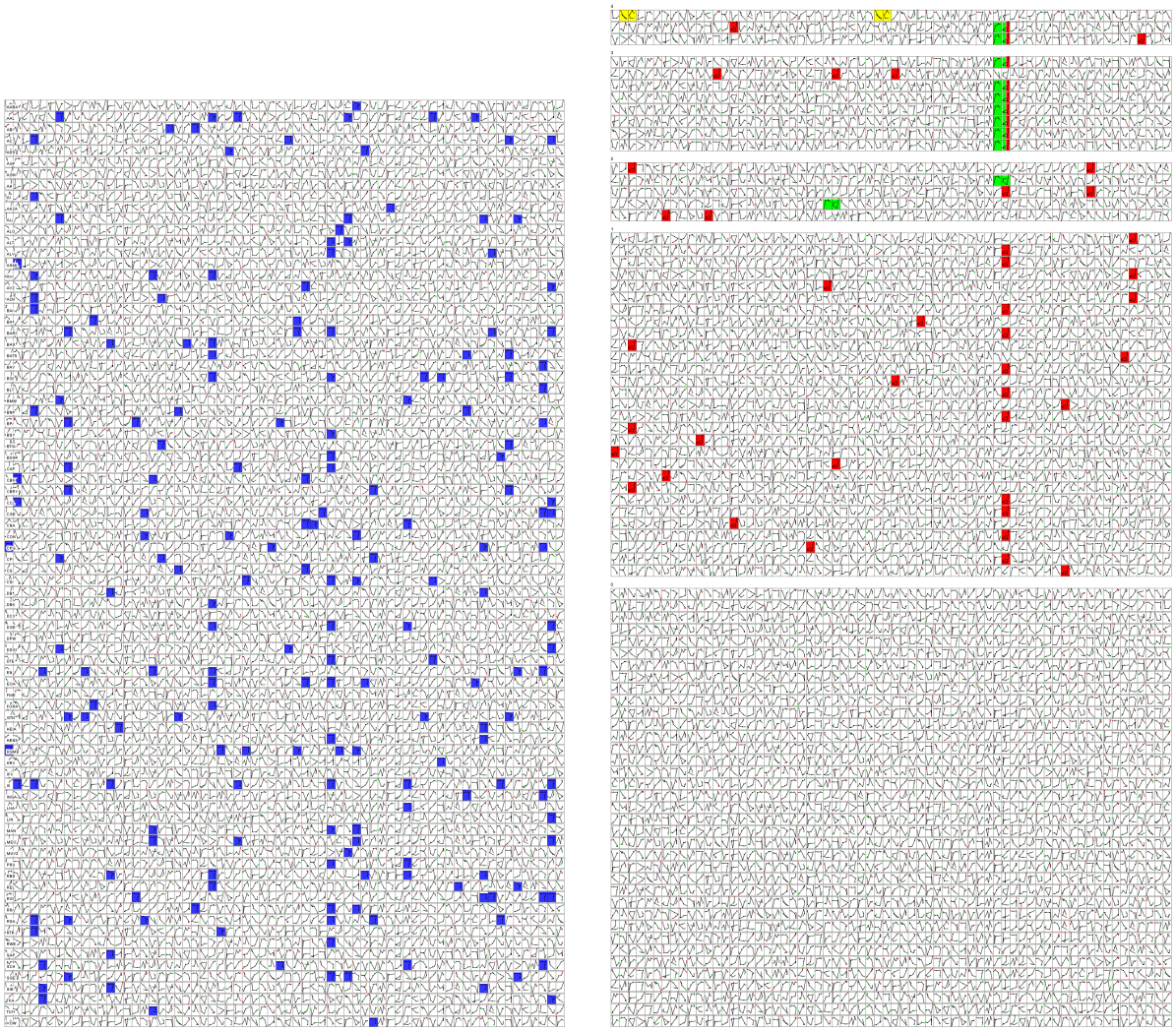


Figure 10.7.: The scatterplot visualizes the dependencies of different, user selected variables. The color represents the length of the displayed subsequence. This visualization helps to find potetially interesting subsequences (e.g., outliers) of subsequences or the correlation of variables.



(a) (b)

Figure 10.8.: Overview of all sequences. a) The most frequent subsequence is highlighted in blue, but no pattern is visible. b) Highlighting of the other three subsequences which have been marked as potentially interesting in the other views during the analysis process. The green and red subsequences occur very often at the same point in time. The yellow subsequence in the first line occurs twice but only in one company. Both findings are starting points for fuhrer financial analysis.

Part III.

Conclusion and Future Work

11. Conclusion and Future Work

It could be shown, that Visual Analytics with its combination of interactive visualizations and data mining can add a valuable application benefit in different areas like biology or evacuation planning. The particular approach and methods applied depend on the respective use case and its data characteristics. I proposed new Visual Analytics solutions for the comparison of descriptors and the explorative analysis of time dependent data.

11.1. Visual Descriptor Space Comparison

Comparing different descriptors of a data set provides inside about its patterns and characteristics. I proposed different scores and visualizations to measure and analyze differences and commonalities for both, data with hierarchic or unknown structure.

Approaches which try to provide the compressed and shared information of many descriptors, are useful for a quick overview or an automated data processing. This work and the feedback of our cooperation partners as well as of other users show, that a detailed comparison of different aspects of the data can lead to valuable insights into the data. Local patterns can be detected and evaluated and the understanding of the detailed structure can enhance the basis of decision making.

My novel tree comparison score takes structural differences of trees into account. Its combination with a visualization tool featuring different linked views for different levels of detail allows for a detailed analysis on the subtree level of many trees simultaneously. The evaluation of the tool in close cooperation with biologists analyzing phylogenetic trees showed the usefulness of my approach. The CloudTree visualization technique extends the scalability of the approach to trees with thousands of leaves. CloudTrees reduce the negative effect of overplotting in the comparative analysis and provide the visibility of even small, but important details.

The evaluation of various state of the art quality measures, originally used for dimensionality reduction, shows their applicability for the comparison of unstructured data. For the visual comparison of two datasets, a two-dimensional color coding is used which can be applied to scatter plots or Self Organizing Maps (SOMs). The SOM comparison visualization combines the dimensionality reduction and clustering capabilities of SOMs with the comparison coloring and uncertainty encoding. In combination with an adaptive comparison pipeline, novice users are guided through the analytical process. The automatic and visual comparison were tested and validated using artificial data. To create these ground truth datasets, I proposed PCDC, an interactive tool for the controlled and detailed creation of high dimensional data.

The combination of the automatic analysis and the interactive visualizations helped the user with the exploration analysis of large datasets. By integrating the user in the analysis, previously unknown patterns could be found and evaluated. The comparison of different representations of the data is a powerful tool, which easily can be extended to other usecases like the parameter steering of algorithms. For example, in the case of the calculation of phylogenetic trees, the biologists were evaluating the influence of the selected representation using a fixed set of parameters for the calculation of the phylogenetic tree. Another interesting question beyond this work is the influence of varying parameters resulting in different trees which could be tackled using my tree comparison

approach. First steps in this direction were recently taken by a master thesis under my supervision by Martin Heß "Interactive Visual Comparison of Many Hierarchic Datasets".

This work adresses the important issue of scalability regarding the number of trees which has not been addressed in this thesis . For the comparison of hierarchies, CloudTrees ensure scalability regarding the size of the trees. The visualization of too many trees at the same time overburdens the user and makes it harder to see details in the respective trees. A promising approach is the clustering of similar trees. I already proposed this successfully for data with unknown structure in the comparison pipeline, where the datasets were hierarchically clustered according to their similarity in terms of neighborhood preservation. For trees, the already established element based similarity measure can be used to form clusters, identify cluster representatives and present them to the users.

Another option for combining tree visualization with the 2D colormap-based approach could be the visualization of hierarchies using treemaps [Shn92b]. Treemaps divide an usually rectangular space according to the hierarchies in sub-rectangles and thereby form a two dimensional space filling projection. Assuming an appropriate layout, the two dimensional colormaps could be applied for their comparison.

Another important topic is the extension of the user evaluation. I already worked in close cooperation with our partners from biology on the tree comparison system without the cloud trees. Since that time, the ViPhy system for the comparison of hierarchies was published and used for research by different biologists worldwide. We already showed examples of value added by the CloudTree extension. A wider user evaluation with domain experts would in future contribute to a better understanding of its advantages and drawback. The same applies to the SOM comparison pipeline. A controlled generation of ground truth test trees with specific sizes and similarity patterns like it was done with PCDC for not hierarchically structured data would be beneficial for an in depth testing of the cloud trees. However, making the tools available to the public as we already did with ViPhy [BvLH*11], Micato & MatrixVis [BSB*10] and PCDC [BHvLF12] is very useful as the feedback from domain experts helps a lot in improving them.

11.2. Analysis of time Series

Visual Analytics provides flexible, usecase specific tools for the analysis of large time dependent data sets. Time Parallel Sets allow for an effective analysis of large sets of categorical data with a focus on class changes of objects. They combine an automatic analysis for the identification of potentially interesting points in time with an interactive visualization. I proposed different algorithms, each specialized on other problems like the identification of representative moments or the detection of outliers. In the detailed visualization of the automatically or manually selected points in time, users have various interactive analysis options like the tracing of single objects or groups. The applicability of the approach has been shown in various application areas like emergency evacuation of buildings and activity tracking of people over long time periods.

The transformation of long time series to sequences can help to compress the data. My approach supports the analyst in the definition of symbol similarity and the analysis of the resulting sequences. Multiple linked views, each specialized on other analytical problems allow for a flexible, task depended analysis. The approach has been applied to financial risk and return estimations of 60 companies over 500 days.

In the future, the scalability of the approaches could be enhanced. In Time Parallel Sets, this applies especially to the number of classes and their ordering as well as for the number of points in time, which is relevant for the sequence analysis as well. The presented approach is suitable mainly for a small number of categories (i.e., small number of category values). Generally, up to about ten categories can be handled in the visualization so that the readability of the views is provided. However, often larger number of categoric values exist in the data set.

This can lead to overplotting problems. In order to support the analysis of a larger number of categoric values, I provide a possibility to interactively merge several categories into one category. For example, the user can merge several similar activities into one broadly-defined activity such as tennis, walking and jogging can be integrated into sports. Alternatively, the user can merge categories that are not in her focus into an "other" category. It allows for targeted analysis of selected categories. As an illustration, Figure 11.1 shows a view on transitions for all categories (left) and for the merged categories (right). The merged view provides a more abstract view of the data with less overploting.

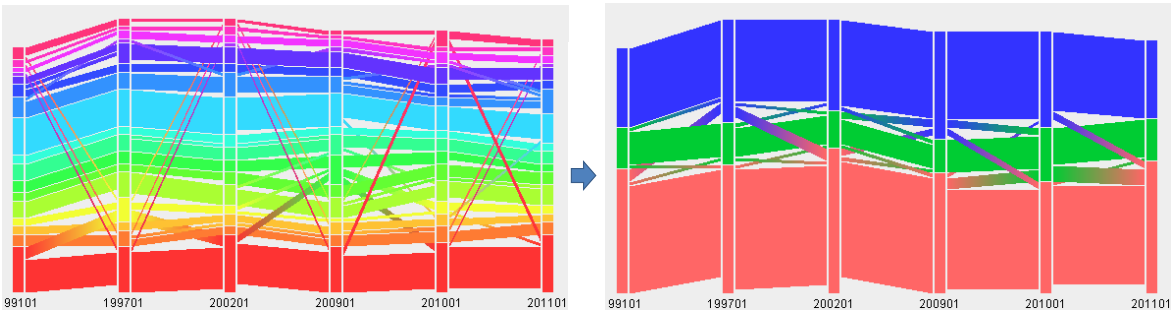


Figure 11.1.: An increasing number of categories can lead to overplotting effects. The merging of similar groups could allow for more scalability.

The ordering of categorical states is important for both, good interpretation and readability of the visualization. Often, this ordering is not pre-defined. We currently provide interactive user-defined ordering. In this case, user experience and hypothesis about possible insights into the data determine the choice of ordering. Alternatively, the ordering can be calculated on the basis of transition counts. For example, methods on the minimization of edge crossings for cluster comparison visualization [LSP*10, ZKG09] or on placement of graph nodes on a line with minimization of edge crossing/edge length [DPS02] can be used. However, these approaches have severe limitations. Finding the optimal solution is NP hard, but moreover, for time dependent data, it is not clear, whether to apply these algorithms on all or only selected time moments. A global minimization may not be optimal for the selected time points and local calculation for selected time points would imply changing the ordering with each time selection change. Therefore, I currently rely on user-defined ordering.

Additionally, the automatic detection of interesting patterns or points in time could be enhanced in the sequence analysis. Right now, for example, the user can select different attributes for the axis in the scatterplot view. Based on algorithms for the evaluation of interestingness of two dimensional point clouds (e.g. [SBS11, LAE*12]), axis combination could be proposed. Additional meta data like the business sector or number of employees in case of financial data, which is actually only used for visualization, could be integrated in the automatic analysis.

The scalability of both approaches, Time Parallel Sets and the sequence analysis, with respect to the number of points in time could be extended by the specific handling of periodic data. Sometimes the periodicity itself is interesting for the analyst, but in other cases, like annual patters in climate research, the periodicity is assumed a priori. In Time Parallel Sets, only transitions varying from these patterns could be used to detect interesting points in time. The determination of differences between multiple periods as well as for the sequences and subsequences could be improved in future. For example, building on the work of Ding et al. [DTS*08], who compared several different distance measurements for the mining of time series could improve the robustness of the identification of common subsequences. As these aspects of similarity measurement are very usecase specific, their influence on the current analysis should be communicated and evaluated by the user.

Part IV.

Appendix

A. Publications

The thesis is partially based on the following publications:

A.1. Journal Publications, Book Chapters

1. VON LANDESBERGER, T., ANDRIENKO, G., ANDRIENKO, N., BREMM, S., KIRSCHNER, M., WESARG, S., KUIJPER, A.:
   Opening up the "Black Box" of Medical Image Segmentation with Statistical Shape Models
   The Visual Computer, 2013, Volume 26, Issue 9, pp 893-905
2. VON LANDESBERGER, T., DIEL, S., BREMM, S., FELLNER, D.W.:
   Visual Analysis of Contagion in Networks
   Information Visualization , 2013, Volume 40, Issue 12, pp. 4934âĂŞ4943
3. VON LANDESBERGER, T., BREMM, S, KIRSCHNER, M., WESARG, S., KUIJPER, A.:
   Visual Analytics for Model-based Medical Image Segmentation: Opportunities and Challenges
   Expert Systems with Applications, Elsevier, 2013
4. LANDESBERGER, T. V., FIEBIG, S., BREMM, S., KUIJPER, A., FELLNER, D.W.:
   Interaction Taxonomy for Tracking of User Actions in Visual Analytics Applications
   Human centric visualization, Springer, 2012
5. TATU, A., ZHANG, L., BERTINI, E., SCHRECK, T., KEIM, D., BREMM, S., VON LANDESBERGER, T.:
   ClustNails: Visual Analysis of Subspace Clusters
   Tsinghua Science and Technology, 2012, Volume 17, Number 4
6. BREMM, S., VON LANDESBERGER, T., BERNARD, J. AND SCHRECK, T.:
   Assisted Descriptor Selection Based on Visual Comparative Data Analysis
   Computer Graphics Forum, 2011, Vol. 30(3), pp. 891-900
7. BREMM, S., SCHRECK, T., BOBA, P., HELD, S. AND HAMACHER, K.:
   Computing and Visually Analyzing Mutual Information in Molecular Co-evolution
   BMC Bioinformatics, 2010, Vol. 11, pp. 330
8. ANDRIENKO, G., ANDRIENKO, N., BREMM, S., SCHRECK, T., LANDESBERGER, T. V., BAK, P. AND KEIM, D.A.:
   Space-in-Time and Time-in-Space Self-Organizing Maps for Exploring Spatiotemporal Patterns
   Computer Graphics Forum, 2010, Vol. 29(3), pp. 913-922
9. SCHRECK, T., VON LANDESBERGER, T. AND BREMM, S.:
   Techniques for Precision-Based Visual Analysis of Projected Data
   Palgrave Macmillan Information Visualization, 2010, Vol. 9(3), pp. 181-193
10. ANDRIENKO, G., ANDRIENKO, N., BAK, P., BREMM, S., KEIM, D.A., LANDESBERGER, T. V., PÖLITZ, C. AND SCHRECK, T.:
    A Framework for Using Self-Organizing Maps to Analyze Spatio-Temporal Patterns,

Exemplified by Analysis of Mobile Phone Usage
Journal of Location Based Services, 2010, Vol. 4(3-4), pp. 200-221

11. BREMM, S., MAIER, S., VON LANDESBERGER, T. AND SCHRECK, T.:
Ein flexibles System für die explorative visuelle Sequenzanalyse
Datenbank Spektrum, 2009, Vol. 31(31), pp. 8-16

A.2. Conference Proceedings

1. VON LANDESBERGER, T., BREMM, S., ANDRIENKO, N. AND ANDRIENKO, G., TEKUSOVA, M.:
Visual Analytics Methods for Categoric Spatio-Temporal Data
IEEE Visual Analytics Science and Technology, 2012
2. VON LANDESBERGER, T., BREMM, S., SCHRECK, T., FELLNER, D.W.:
Feature-based Identification of Interesting Data Segments in Group Movement Data
GIScience workshop on GeoVisual Analytics , 2012
3. BREMM, S. , VON LANDESBERGER, T., HESS, M. AND FELLNER, D.W.:
PCDC - On the Highway to Data - A Tool for the Fast Generation of Large Synthetic Data Sets
EuroVA International Workshop on Visual Analytics, 2012
4. BREMM, S. , VON LANDESBERGER, T., HESS, M., SCHRECK, T., WEIL, P. AND HAMACHER, K. :
Interactive Visual Comparison of Multiple Trees
IEEE Visual Analytics Science and Technology, 2011, pp. 31âĂŞ40
5. BREMM, S., VON LANDESBERGER, T., ANDRIENKO, G. AND ANDRIENKO, N.:
Interactive Analysis of Object Group Changes over Time
EuroVA International Workshop on Visual Analytics, 2011, pp. 41-44
6. BERNARD, J., LANDESBERGER, T. V., BREMM, S. AND SCHRECK, T.:
Multi-Scale Visual Quality Assessment for Cluster Analysis with Self-Organizing Maps
Visualization and Data Analysis 2011 SPIE Press, 2011, pp. 78680N-1-78680N-12
7. LANDESBERGER, T. V., BREMM, S., BERNARD, J. AND SCHRECK, T.:
Smart Query Definition for Content-Based Search in Large Sets of Graphs
EuroVAST 2010 Eurographics Association, 2010, pp. 7-12
8. SCHRECK, T., VON LANDESBERGER, T. AND BREMM, S.:
Techniques for Precision-Based Visual Analysis of Projected Data
Proceedings of IS&T/SPIE Electronic Imaging: Visualization and Data Analysis, 2010, Vol. 7530, pp. 75300E-75300E-12
9. VON LANDESBERGER, T., BREMM, S., REZAEI, P. AND SCHRECK, T.:
Visual Analytics of Time Dependent 2D Point Clouds
Computer Graphics International, 2009, pp. 97-101

B. Supervising Activities

The following list summarizes the student bachelor, diploma and master thesis supervised by the author. The results of these works were partially used as an input into the thesis.

B.1. Diploma and Master Thesis

1. Sebastian Meier: Interactive Analyse von Zeitserien, 2009
2. Martin Heß: Interactive Visual Comparison of Many Hierarchic Datasets, 2013

B.2. Bachelor Thesis

1. Martin Heß: Visueller Vergleich hierarchisch organisierter Daten, 2011

C. Curriculum Vitae

Personal Data

Name	Sebastian Bremm
Birth date & place	12.08.1981 Frankfurt a.M.
Family status	Married, one child
Nationality	German

Education

2008	Graduation in Bioinformatics at the J.W. Goethe University in Frankfurt, Germany
2007 – 2008	Diploma Thesis „Interaktive Visualisierung von hochdimensionalen Daten im Bereich der Chemoinformatik“ at Merck-Serono, Darmstadt, Germany
2002 – 2008	Study of Bioinformatics at the J.W. Goethe University in Frankfurt, Germany

Work Experience

2008 – 2013	Researcher, Interactive Graphics Systems Group, Technische Universität Darmstadt, Germany, Focus: Visual Descriptor Space Comparison and Visual Analysis of Time Series
2008	Database development at the Laboratory for Stem Cell Transplantation and Immunotherapy, Johann Wolfgang Goethe-University Hospital, Frankfurt, Germany
2007	Student trainee at Sanofi-Aventis, Frankfurt & Bad Soden, Germany

D. List of Abbreviations

CIE	Commission Internationale de l'Éclairage (International Commission on Illumination)
HTS	High Throughput Screening
LCMC	Local Continuity Metacriterion
MDS	Multidimensional Scaling
MRRE	Mean Relative Rank Errors
PCA	Principal Component Analysis
PCDC	Parallel Coordinates Data Creator
PP	Projection Precision
RGB	Red Green Blue
SOM	Self-Organizing Map
T&C	Trustworthiness and Continuity
TIPS	Time Parallel Sets
VA	Visual Analytics

Bibliography

[AA10] ANDRIENKO G., ANDRIENKO N.: Interactive cluster analysis of diverse types of spatio-temporal data. *SIGKDD Explor. Newsl. 11* (May 2010), 19–28. viii, 88

[AG06] ANISIMOVA M., GASCUEL O.: Approximate likelihood-ratio test for branches: A fast, accurate, and powerful alternative. *Systematic Biology 55*, 4 (Aug. 2006), 539–552. PMID: 16785212. 29

[AGM*90] ALTSCHUL S. F., GISH W., MILLER W., MYERS E. W., LIPMAN D. J.: Basic local alignment search tool. *Journal of molecular biology 215*, 3 (1990), 403–410. 95

[AHK01] AGGARWAL C., HINNEBURG A., KEIM D.: On the surprising behavior of distance metrics in high dimensional space. *Database Theory, ICDT 2001 1973* (2001), 420–434. 65

[AMM*08] AIGNER W., MIKSCH S., MULLER W., SCHUMANN H., TOMINSKI C.: Visual methods for analyzing time-oriented data. *IEEE Transactions on Visualization and Computer Graphics 14*, 1 (2008), 47 –60. 78

[AMST11] AIGNER W., MIKSCH S., SCHUMANN H., TOMINSKI C.: *Visualization of time-oriented data*. Springer-Verlag New York Inc, 2011. 78

[BBG*09] BLAAS J., BOTHA C., GRUNDY E., JONES M., LARAMEE R., POST F.: Smooth graphs for visual exploration of higher-order state transitions. *Visualization and Computer Graphics, IEEE Transactions on 15*, 6 (2009), 969 –976. 78

[BD07] BALZER M., DEUSSEN O.: Level-of-detail visualization of clustered graph layouts. In *Visualization, 2007. APVIS'07. 2007 6th International Asia-Pacific Symposium on* (2007), Ieee, pp. 133–140. 33

[BGRS99] BEYER K., GOLDSTEIN J., RAMAKRISHNAN R., SHAFT U.: When is nearest neighbor meaningful? *Database Theory 1540/1999* (1999), 217–235. 65

[BHvLF12] BREMM S., HESS M., VON LANDESBERGER T., FELLNER D.: Pcdc-on the highway to data-a tool for the fast generation of large synthetic data sets. In *EuroVA 2012: International Workshop on Visual Analytics* (2012), The Eurographics Association, pp. 7–11. 49, 50, 108

[BKH*11] BURCH M., KONEVTSOVA N., HEINRICH J., HOEFERLIN M., WEISKOPF D.: Evaluation of traditional, orthogonal, and radial tree diagrams by an eye tracking study. *Visualization and Computer Graphics, IEEE Transactions on 17*, 12 (2011), 2440–2448. 22, 34

[BMvLS09] BREMM S., MAIER S., VON LANDESBERGER T., SCHRECK T.: Ein flexibles system für die explorative visuelle Sequenzanalyse. *Datenbank Spektrum 31*, 31 (December 2009), 8–16. 95

[Bre12] BREWER C. A.: ColorBrewer, 2012. www.colorbrewer2.org, accessed 4.7.2012. 22

[BSB*10] BREMM S., SCHRECK T., BOBA P., HELD S., HAMACHER K.: Computing and visually analyzing mutual information in molecular co-evolution. *BMC Bioinformatics 11* (2010), 330. 108

[BST04] BÖCKER A., SCHNEIDER G., TECKENTRUP A.: Status of HTS data mining approaches. *QSAR & combinatorial science 23*, 4 (2004), 207–213. 70

[BvLAA11] BREMM S., VON LANDESBERGER T., ANDRIENKO G., ANDRIENKO N. .: Interactive analysis of object group changes over time. In *EuroVA 2011, Second International Workshop on Visual Analytics* (2011), pp. 41–44. 77

[BvLBS11] BREMM S., VON LANDESBERGER T., BERNARD J., SCHRECK T.: Assisted descriptor selection based on visual comparative data analysis. *Computer Graphics Forum 30*, 3 (2011), 891–900. 49, 65

[BvLH*] BREMM S., VON LANDESBERGER T., HEÃ§ M., TOBIAS SCHRECK P. W., HAMACHER K.: Viphy - interactive visual comparison of multiple phylogenetic trees. Website. 30

[BvLH*11] BREMM S., VON LANDESBERGER T., HESS M., SCHRECK T., WEIL P., HAMACHER K.: Interactive visual comparison of multiple trees. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on* (2011), IEEE, pp. 31–40. 10, 16, 108

[CAC*09] COCK P. J. A., ANTAO T., CHANG J. T., CHAPMAN B. A., COX C. J., DALKE A., FRIEDBERG I., HAMELRYCK T., KAUFF F., WILCZYNSKI B., DE HOON M. J. L.: Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics (Oxford, England) 25*, 11 (June 2009), 1422–1423. PMID: 19304878. 29

[Cas00] CASTRESANA J.: Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. *Molecular Biology and Evolution 17*, 4 (Apr. 2000), 540–552. PMID: 10742046. 29

[CB09] CHEN L., BUJA A.: Local multidimensional scaling for nonlinear dimension reduction, graph drawing, and proximity analysis. *Journal of the American Statistical Association 104*, 485 (2009), 209–219. 53

[CFC*01] CHIU T., FANG D., CHEN J., WANG Y., JERIS C.: A robust and scalable clustering algorithm for mixed type attributes in large database environment. In *Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining* (2001), ACM, pp. 263–268. 67

[CK95] CARRIERE J., KAZMAN R.: Research report. interacting with huge hierarchies: beyond cone trees. In *Information Visualization, 1995. Proceedings.* (1995), IEEE, pp. 74–81. 33

[CLKP10] CHOO J., LEE H., KIHM J., PARK H.: iVisClassifier: An Interactive Visual Analytics System for Classification Based on Supervised Dimension Reduction. In *IEEE Symposium on Visual Analytics Science and Technology* (2010), pp. 27–34. 65

[CMP09] CHEN J., MACEACHREN A., PEUQUET D.: Constructing Overview+ Detail Dendrogram-Matrix Views. *Visualization and Computer Graphics, IEEE Transactions on 15*, 6 (2009), 889–896. 11, 26, 34

[CMS99] CARD S. K., MACKINLAY J. D., SHNEIDERMAN B.: *Readings in information visualization: using vision to think*. Morgan Kaufmann Pub, 1999. iii, 1

[CN02] CARD S. K., NATION D.: Degree-of-interest trees: a component of an attention-reactive user interface. In *AVI '02: Proceedings of the Working Conference on Advanced Visual Interfaces* (New York, NY, USA, 2002), ACM, pp. 231–245. 11, 33

[CSK*03] CHENNA R., SUGAWARA H., KOIKE T., LOPEZ R., GIBSON T. J., HIGGINS D. G., THOMPSON J. D.: Multiple sequence alignment with the clustal series of programs. *Nucleic acids research 31*, 13 (2003), 3497–3500. 95

[DCSL02] DASH M., CHOI K., SCHEUERMANN P., LIU H.: Feature selection for clustering-a filter solution. In *Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on* (2002), IEEE, pp. 115–122. 49

[DF84] DIACONIS P., FREEDMAN D.: Asymptotics of graphical projection pursuit. *The annals of statistics 12* (1984), 793–815. 49

[DGB*08] DEREEPER A., GUIGNON V., BLANC G., AUDIC S., BUFFET S., CHEVENET F., DUFAYARD J., GUINDON S., LEFORT V., LESCOT M., CLAVERIE J., GASCUEL O.: Phylogeny.fr: robust phylogenetic analysis for the non-specialist. *Nucleic Acids Research 36*, Web Server (2008), W465–W469. 10, 22, 29

[DLH11] DAAE LAMPE O., HAUSER H.: Interactive visualization of streaming data with kernel density estimation. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE* (2011), IEEE, pp. 171–178. 34

[DPS02] DÍAZ J., PETIT J., SERNA M.: A survey of graph layout problems. *ACM Comput. Surv. 34* (Sep. 2002), 313–356. 109

[DTS*08] DING H., TRAJCEVSKI G., SCHEUERMANN P., WANG X., KEOGH E.: Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment 1*, 2 (2008), 1542–1552. 109

[Edg04] EDGAR R. C.: MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research 32*, 5 (Mar. 2004), 1792 –1797. 29

[EF10] ELMQVIST N., FEKETE J.: Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines. *Visualization and Computer Graphics, IEEE Transactions on 16*, 3 (2010), 439–454. 11, 34

[FPSG10] FREIRE M., PLAISANT C., SHNEIDERMAN B., GOLBECK J.: ManyNets: an interface for multiple network analysis and visualization. In *Proceedings of international conference on Human factors in computing systems* (New York, NY, USA, 2010), ACM, pp. 213–222. 13

[Fro05] FROEHLICH P.: Interactive Sankey Diagrams. In *IEEE Symp. on Information Visualization* (2005), p. 233. 77

[FTM*07] FINN R. D., TATE J., MISTRY J., COGGILL P. C., SAMMUT S. J., HOTZ H., CERIC G., FORSLUND K., EDDY S. R., SONNHAMMER E. L. L., BATEMAN A.: The pfam protein families database. *Nucleic Acids Research 36*, Database (2007), D281–D288. 29

[Fuk90] FUKUNAGA K.: *Introduction to statistical pattern recognition*. Academic Pr, 1990. 65

[GCML06] GUO D., CHEN J., MACEACHREN A., LIAO K.: A visualization system for space-time and multivariate patterns (vis-stamp). *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (2006), 1461–1474. 78

[GE03] GUYON I., ELISSEEFF A.: An introduction to variable and feature selection. *The Journal of Machine Learning Research 3* (2003), 1157–1182. 49

[GG03] GUINDON S., GASCUEL O.: A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Systematic Biology 52*, 5 (Oct. 2003), 696–704. PMID: 14530136. 29

[GK09] GRAHAM M., KENNEDY J.: A survey of multiple tree visualisation. *Information Visualization 9*, 4 (2009), 235–252. 11

[GPL*08] GRINSTEIN G., PLAISANT C., LASKOWSKI S., O'CONNELL T., SCHOLTZ J., WHITING M.: VAST 2008 Challenge: Introducing mini-challenges. In *IEEE Symp. on Visual Analytics*

Science and Technology (2008), pp. 195–196. 87, 88

[Hal99] HALL M. A.: *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999. 49

[Ham10] HAMACHER K.: Protein domain phylogenies - information theory and evolutionary dynamics. In *Bioinformatics* (2010), Fred A., Filipe J., Gamboa H., (Eds.), pp. 114–122. 9, 28

[HBB*08] HULO N., BAIROCH A., BULLIARD V., CERUTTI L., CUCHE B. A., DE CASTRO E., LACHAIZE C., LANGENDIJK-GENEVAUX P. S., SIGRIST C. J. A.: The 20 years of PROSITE. *Nucleic Acids Research 36*, Database issue (Jan. 2008), D245–D249. PMID: 18003654 PMCID: 2238851. 30

[HC04] HEER J., CARD S. K.: Doitrees revisited: scalable, space-constrained visualization of hierarchical data. In *AVI '04: Proceedings of the working conference on Advanced visual interfaces* (New York, NY, USA, 2004), ACM, pp. 421–424. 11, 33

[HCN06] HE X., CAI D., NIYOGI P.: Laplacian score for feature selection. *Advances in Neural Information Processing Systems 18* (2006), 507. 49

[HDKS07] HAO M. C., DAYAL U., KEIM D. A., SCHRECK T.: Multi-resolution techniques for visual exploration of large time-series data. In *EuroVis* (2007), pp. 27–34. 78

[HDM98] HERMAN I., DELEST M., MELANCON G.: Tree visualisation and navigation clues for information visualisation. In *Computer Graphics Forum* (1998), vol. 17, Wiley Online Library, pp. 153–165. 34

[HH92] HENIKOFF S., HENIKOFF J.: Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences 89*, 22 (1992), 10915–10919. 96

[HHJ05] HILLIS D., HEATH T., JOHN K.: Analysis and visualization of tree space. *Systematic Biology 54*, 3 (2005), 471. 13

[HHN02] HAVRE S., HETZLER B., NOWELL L.: ThemeRiver: Visualizing theme changes over time. In *IEEE Symp. on Information Visualization* (2002), pp. 115–123. 77, 78

[Him98] HIMBERG J.: Enhancing SOM-based data visualization by linking different data projections. In *Int. Symp. on Intelligent Data Engineering and Learning* (1998), Eureka, p. 427. 54

[Him00] HIMBERG J.: A SOM based cluster visualization and its application for false coloring. In *IEEE Int. Joint Conf. on Neural Networks* (2000), vol. 3, p. 3587. 54, 57

[HJM*11] HAO M., JANETZKO H., MITTELSTÄDT S., HILL W., DAYAL U., KEIM D., MARWAH M., SHARMA R.: A visual analytics approach for peak-preserving prediction of large seasonal time series. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 691–700. 79

[HMM00] HERMAN I., MELANÇON G., MARSHALL M. S.: Graph visualization and navigation in information visualization: A survey. *Visualization and Computer Graphics, IEEE Transactions on 6*, 1 (2000), 24–43. 33

[Hof08] HOFMANN H.: Mosaic plots and their variants. *Handbook of data visualization* (2008), 617–642. 77

[HRR*07] HUSON D., RICHTER D., RAUSCH C., DEZULIAN T., FRANZ M., RUPP R.: Dendroscope: An interactive viewer for large phylogenetic trees. *Bmc Bioinformatics 8*, 1 (2007), 460. 10

[HVW08] HOLTEN D., VAN WIJK J.: Visual comparison of hierarchically organized data. *Computer Graphics Forum 27*, 3 (2008), 759–766. 11

[IMI*10] INGRAM S., MUNZNER T., IRVINE V., TORY M., BERGNER S., MÖLLER T.: DimStiller: Workflows for dimensional analysis and reduction. In *IEEE Conference on Visual Analytics Software and Technologies* (2010), pp. 3–10. 66

[Ins] INSTITUTE OF EVOLUTIONARY BIOLOGY, UNIVERSITY OF EDINBURGH: Figtree. http://tree.bio.ed.ac.uk/software/figtree/. 9

[Ins85] INSELBERG A.: The plane with parallel coordinates. *The Visual Computer 1*, 2 (1985), 69–91. 77

[JJ09] JOHANSSON S., JOHANSSON J.: Interactive dimensionality reduction through user-defined combinations of quality metrics. *Visualization and Computer Graphics, IEEE Transactions on 15*, 6 (2009), 993–1000. 66, 87

[KBH06] KOSARA R., BENDIX F., HAUSER H.: Parallel sets: Interactive exploration and visual analysis of categorical data. *Visualization and Computer Graphics, IEEE Transactions on 12*, 4 (2006), 558–568. viii, 77, 78, 79

[KBK11] KRSTAJIC M., BERTINI E., KEIM D.: Cloudlines: compact display of event episodes in multiple time-series. *Visualization and Computer Graphics, IEEE Transactions on 17*, 12 (2011), 2432–2439. 34

[KFS13] KOOP D., FREIRE J., SILVA C. T.: Visual summaries for graph collections. In *IEEE PacificVis* (2013). 12

[KK98] KASKI S., KOHONEN T.: *Visual Explorations in Finance*. Springer, 1998, ch. Tips for processing and color-coding of Self-Organizing Maps, pp. 195–202. 54

[KKEM10] KEIM D. A., KOHLHAMMER J., ELLIS G., MANSMANN F.: *Mastering The Information Age-Solving Problems with Visual Analytics*. Florian Mansmann, 2010. 1, 2

[KKZ09] KRIEGEL H., KRÖGER P., ZIMEK A.: Outlier detection techniques. In *Tutorial at the 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining* (2009). 87

[Koh82] KOHONEN T.: Self-organized formation of topologically correct feature maps. *Biological cybernetics 43*, 1 (1982), 59–69. 50

[Koh01] KOHONEN T.: *Self-Organizing Maps*. Springer, 2001. 96

[KS96] KETCHEN D., SHOOK C.: The application of cluster analysis in strategic management research: an analysis and critique. *Strategic management journal 17*, 6 (1996), 441–458. 87

[KVK99] KASKI S., VENNA J., KOHONEN T.: Coloring that reveals high-dimensional structures in data. In *Int. Conf. on Neural Information Processing* (1999), vol. 2, pp. 729 – 734. 54

[KVK00] KASKI S., VENNA J., KOHONEN T.: Coloring that reveals cluster structures in multivariate data. *Australian J. of Intelligent Information Processing Systems 6*, 2 (2000), 82–88. 54

[LAE*12] LEHMANN D. J., ALBUQUERQUE G., EISEMANN M., MAGNOR M., THEISEL H.: Selecting coherent and relevant plots in large scatterplot matrices. *Computer Graphics Forum 31*, 6 (Apr. 2012), 1895–1908. 109

[LB06] LETUNIC I., BORK P.: Interactive tree of life (itol): an online tool for phylogenetic tree display and annotation. *Bioinformatics 23*, 1 (2006), 127. 10

[LBX*01] LI M., BADGER J., XIN C., KWONG S., KEARNEY P. E.: An information based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics 17* (2001), 149–154. 9, 28

[LG07] LEACH A., GILLET V.: *An introduction to chemoinformatics*. Springer Verlag, 2007. 43

[LH03] LIN Y.-L., HSU T.-S.: Efficient algorithms for descendent subtrees comparison of phylogenetic trees with applications to co-evolutionary classifications in bacterial genome. In *Algorithms and Computation*, Ibaraki T., Katoh N., Ono H., (Eds.), vol. 2906 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, 2003, pp. 339–351. 21

[LH11] LAMPE O., HAUSER H.: Curve density estimates. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 633–642. 79

[LKL05] LIN J., KEOGH E., LONARDI S.: Visualizing and discovering non-trivial patterns in large time series databases. *Information visualization 4*, 2 (2005), 61–82. 95

[LKLC03] LIN J., KEOGH E., LONARDI S., CHIU B.: A symbolic representation of time series, with implications for streaming algorithms. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery* (2003), ACM, pp. 2–11. 95

[LLW10] LIU K., LINDER C., WARNOW T.: Multiple sequence alignment: a major challenge to large-scale phylogenetics. *PLoS currents 3* (2010), RRN1198. 45

[LSP*10] LEX A., STREIT M., PARTL C., KASHOFER K., SCHMALSTIEG D.: Comparative analysis of multidimensional, quantitative data. *IEEE Trans. on Visualization and Computer Graphics 16*, 6 (11 2010), 1027–1035. 77, 78, 109

[LV07] LEE J., VERLEYSEN M.: *Nonlinear dimensionality reduction*. Springer, 2007. 53

[LV09] LEE J., VERLEYSEN M.: Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing 72*, 7 (2009), 1431–1443. 53, 54

[LY07] LIN C.-C., YEN H.-C.: On balloon drawings of rooted trees. *J. Graph Algorithms Appl. 11*, 2 (2007), 431–452. 33

[May07] MAY T.: Working with patterns in large multivariate datasets-karnaugh-veitch-maps revisited. In *11th Int. Conf. on Information Visualization* (2007), IEEE, pp. 277–285. 77

[MGT*03] MUNZNER T., GUIMBRETIÈRE F., TASIRAN S., ZHANG L., ZHOU Y.: Treejuxtaposer: scalable tree comparison using focus+context with guaranteed visibility. *ACM Trans. Graph. 22* (July 2003), 453–462. 11, 12, 15, 16, 21, 24, 33, 34

[NCA06] NEUMANN P., CARPENDALE M. S. T., AGARAWALA A.: Phyllotrees: Phyllotactic patterns for tree layout. In *Proceedings of the Joint Eurographics and IEEE TCVG Symposium on Visualization* (2006), Eurographics, pp. 59–66. 33

[NH03] NGUYEN Q. V., HUANG M. L.: Space-optimized tree: a connection+ enclosure approach for the visualization of large hierarchies. *Information Visualization 2*, 1 (2003), 3–15. 33

[Nye08] NYE T.: Trees of trees: an approach to comparing multiple alternative phylogenies. *Systematic biology 57*, 5 (2008), 785. 13

[OHS05] O'ROURKE N., HATCHER L., STEPANSKI E. J.: *A step-by-step approach to using SAS for univariate and multivariate statistics*. SAS Press, 2005. 49

[PDT09] PANAGIOTAKIS C., DOULAMIS A., TZIRITAS G.: Equivalent key frames selection based on iso-content principles. *Circuits and Systems for Video Technology, IEEE Transactions on 19*, 3 (2009), 447–451. 79

[PGM07] PUIGBO P., GARCIA-VALLE S., MCINERNEY J. O.: TOPD/FMTS: a new software to compare phylogenetic trees. *Bioinformatics 23*, 12 (June 2007), 1556 –1558. 12

[PHH10] PAPE S., HOFFGAARD F., HAMACHER K.: Distance-dependent classification of amino acids by information theory. *Proteins: Structure, Function, and Bioinformatics 78*, 10 (2010),

2322–2328. 13

[RMC91] ROBERTSON G. G., MACKINLAY J. D., CARD S. K.: Cone trees: animated 3d visualizations of hierarchical information. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (1991), ACM, pp. 189–194. 33

[RT60] ROGERS D. J., TANIMOTO T. T.: A computer program for classifying plants. *Science 132*, 3434 (1960), 1115–1118. 43

[SBS11] SCHERER M., BERNARD J., SCHRECK T.: Retrieval and exploratory search in multivariate research data repositories using regressional features. In *Proceedings of the 11th annual international ACM/IEEE joint conference on Digital libraries* (2011), ACM, pp. 363–372. 109

[SBVLK09] SCHRECK T., BERNARD J., VON LANDESBERGER T., KOHLHAMMER J.: Visual cluster analysis of trajectory data with interactive kohonen maps. *Information Visualization 8*, 1 (2009), 14–29. 95, 96

[SC04] SALVADOR S., CHAN P.: Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms. In *Tools with Artificial Intelligence, 2004. ICTAI 2004. 16th IEEE International Conference on* (2004), IEEE, pp. 576–584. 67

[Shn92a] SHNEIDERMAN B.: Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on graphics (TOG) 11*, 1 (1992), 92–99. 10

[Shn92b] SHNEIDERMAN B.: Tree visualization with tree-maps: 2-d space-filling approach. *ACM Transactions on graphics (TOG) 11*, 1 (1992), 92–99. 108

[Shn96] SHNEIDERMAN B.: The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on* (1996), IEEE, pp. 336–343. iii, 1, 33

[SK08] SHIEH J., KEOGH E.: i sax: indexing and mining terabyte sized time series. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (2008), ACM, pp. 623–631. 95

[SP93] STEEL M., PENNY D.: Distributions of tree comparison metrics–some new results. *Systematic Biology 42*, 2 (1993), 126–141. 15, 16, 21

[SS02] SEO J., SHNEIDERMAN B.: Interactively exploring hierarchical clustering results [gene identification]. *Computer 35*, 7 (2002), 80–86. 34

[SSH09] SCHULZ H.-J., SCHUMANN H., HADLAK S.: Point-based tree representation - a new approach for large hierarchies. In *Proceedings of IEEE Pacific Visualization Symposium* (2009), pp. 81–88. 33, 34

[STKF07] SCHRECK T., TEKUŠOVÁ T., KOHLHAMMER J., FELLNER D.: Trajectory-based visual analysis of large financial time series data. *ACM SIGKDD Explorations Newsletter 9*, 2 (2007), 30–37. 96

[Str57] STRAHLER A.: Quantitative analysis of watershed geomorphology. *Transactions of the American Geophysical Union 38*, 6 (1957), 913–920. 34, 39

[SvLB10] SCHRECK T., VON LANDESBERGER T., BREMM S.: Techniques for precision-based visual analysis of projected data. *Palgrave Macmillan Information Visualization 9*, 3 (2010), 181–193. 49, 53, 58

[SW49] SHANNON C. E., WEAVER W.: The mathematical theory of communication (urbana, il. *University of Illinois Press 19*, 7 (1949), 1. 49

[TA08] TELEA A., AUBER D.: Code flows: Visualizing structural evolution of source code. In *Computer Graphics Forum* (2008), vol. 27, Wiley Online Library, pp. 831–838. 12

[TAE*09] TATU A., ALBUQUERQUE G., EISEMANN M., SCHNEIDEWIND J., THEISEL H., MAGNOR M., KEIM D.: Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *IEEE Symp. on Visual Analytics Science andTechnology* (2009), pp. 59–66. 66

[TAvHS06] TOMINSKI C., ABELLO J., VAN HAM F., SCHUMANN H.: Fisheye tree views and lenses for graph visualization. In *Proceedings of IEEE Information Visualization* (2006), pp. 17–24. 33

[TC05] THOMAS J. J., COOK K. A.: *Illuminating the path: The research and development agenda for visual analytics*. IEEE Computer Society Press, 2005. iii, 1

[TH83] TUFTE E., HOWARD G.: *The visual display of quantitative information*, vol. 7. Graphics press Cheshire, CT, USA, 1983. 78

[TKL02] TEOH S. T., KWAN-LIU M.: Rings: A technique for visualizing large hierarchies. In *Graph Drawing* (2002), Springer, pp. 268–275. 33

[TPRH11] TURKAY C., PARULEK J., REUTER N., HAUSER H.: Interactive visual analysis of temporal cluster structures. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 711–720. 78

[TS07] TU Y., SHEN H.: Visualizing changes of hierarchical data using treemaps. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1286–1293. 11, 12

[TS08] TU Y., SHEN H.-W.: Balloon focus: a seamless multi-focus+context method for treemaps. *Visualization and Computer Graphics, IEEE Transactions on 14*, 6 (Nov.-Dec. 2008), 1157–1164. 11, 33

[Tuk77] TUKEY J. W.: Exploratory data analysis, 1977. iii, 1

[TV07] TRUONG B., VENKATESH S.: Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP) 3*, 1 (2007), 3. 79

[TW05] TIBSHIRANI R., WALTHER G.: Cluster validation by prediction strength. *Journal of Computational and Graphical Statistics 14*, 3 (2005), 511–528. 67

[TWH01] TIBSHIRANI R., WALTHER G., HASTIE T.: Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63*, 2 (2001), 411–423. 67

[VA00] VESANTO J., ALHONIEMI E.: Clustering of the self-organizing map. *Trans. on Neural Networks 11*, 3 (2000), 586–600. 61

[vHvdWvW01] VAN HAM F., VAN DE WETERING H., VAN WIJK J.: Visualization of state transition graphs. In *Information Visualization, 2001. INFOVIS 2001. IEEE Symposium on* (2001), pp. 59 –66. 78

[VJC09] VROTSOU K., JOHANSSON J., COOPER M.: Activitree: Interactive visual exploration of sequences in event-based data using graph similarity. *Visualization and Computer Graphics, IEEE Transactions on 15*, 6 (nov.-dec. 2009), 945 –952. 78

[VK01] VENNA J., KASKI S.: Neighborhood preservation in nonlinear projection methods: An experimental study. *Artificial Neural Networks, ICANN 2001 2130* (2001), 485–491. 53

[vLBA*12] VON LANDESBERGER T., BREMM S., ANDRIENKO N., ANDRIENKO G., TEKUSOVA M.: Visual analytics methods for categoric spatio-temporal data. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on* (2012), IEEE, pp. 183–192. 77

[vLGS09] VON LANDESBERGER T., GÖRNER M., SCHRECK T.: Visual analysis of graphs with multiple connected components. In *Proceedings of IEEE Symposium on Visual Analytics Science and Technology* (2009), pp. 155–162. 13

[vLKS*11] VON LANDESBERGER T., KUIJPER A., SCHRECK T., KOHLHAMMER J., VAN WIJK J., FEKETE J.-D., FELLNER D.: Visual analysis of large graphs: State-of-the-art and future research challenges. *Computer Graphics Forum 30*, 6 (2011), 1719–1749. 10

[VWD04] VIEGAS F. B., WATTENBERG M., DAVE K.: Studying cooperation and conflict between authors studying cooperation and conflict between authors. In *CHI* (2004), pp. 575–582. 78

[WBDS11] WOOD J., BADAWOOD D., DYKES J., SLINGSBY A.: Ballotmaps: Detecting name bias in alphabetically ordered ballot papers. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2384–2391. 78

[WCF*00] WONG P. C., COWLEY W., FOOTE H., JURRUS E., THOMAS J.: Visualizing sequential patterns for text mining. In *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on* (2000), IEEE, pp. 105–111. 95, 97

[WD08] WOOD J., DYKES J.: Spatially ordered treemaps. *Visualization and Computer Graphics, IEEE Transactions on 14*, 6 (2008), 1348–1355. 78

[WFH11] WITTEN I., FRANK E., HALL M.: *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2011. 85, 87

[WG11] WARD M. O., GUO Z.: Visual exploration of time-series data with shape space projections. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 701–710. 95

[WHM*09] WU D., HUGENHOLTZ P., MAVROMATIS K., PUKALL R., DALIN E., IVANOVA N. N., KUNIN V., GOODWIN L., WU M., TINDALL B. J., HOOPER S. D., PATI A., LYKIDIS A., SPRING S., ANDERSON I. J., D/'HAESELEER P., ZEMLA A., SINGER M., LAPIDUS A., NOLAN M., COPELAND A., HAN C., CHEN F., CHENG J.-F., LUCAS S., KERFELD C., LANG E., GRONOW S., CHAIN P., BRUCE D., RUBIN E. M., KYRPIDES N. C., KLENK H.-P., EISEN J. A.: A phylogeny-driven genomic encyclopaedia of bacteria and archaea. *Nature 462* (2009), 1056 – 1060. 29

[WS06] WEINBERGER K. Q., SAUL L. K.: An introduction to nonlinear dimensionality reduction by maximum variance unfolding. In *Proceedings of the National Conference on Artificial Intelligence* (2006), vol. 21, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, p. 1683. 50

[ZJGK10] ZIEGLER H., JENNY M., GRUSE T., KEIM D.: Visual market sector analysis for financial time series data. In *IEEE Symposium on Visual Analytics Science and Technology* (2010), pp. 83 –90. 79

[ZKG09] ZHOU J., KONECNI S., GRINSTEIN G.: Visually comparing multiple partitions of data with applications to clustering. In *Proceedings of SPIE* (2009), vol. 7243, p. 72430J. 77, 78, 109

[ZMP97] ZHONG Y., MEACHAM C., PRAMANIK S.: A general method for tree-comparison based on subtree similarity and its use in a taxonomic database. *Biosystems 42*, 1 (1997), 1–8. 15

[ZNK08] ZIEGLER H., NIETZSCHMANN T., KEIM D.: Visual analytics on the financial market: Pixel-based analysis and comparison of long-term investments. In *International Conference on*

Information Visualisation (2008), pp. 287 –295. 78, 79